Computational Approaches to Construction Grammar and Morphology

Inaugural-Dissertation
zur Erlangung des Doktorgrades der Philosophie
an der Ludwig–Maximilians–Universität München

vorgelegt von
Leonie Alexandra Weißweiler
aus Herdecke

2025

Referent: Prof. Dr. Hinrich Schütze

Korreferentin: Prof. Dr. Barbara Plank

Tag der Einreichung: 14. März 2024
Tag der mündlichen Prüfung: 03. Juli 2024

Abstract

For the past 100 years, there has been a debate in Linguistics and Natural Language Processing (NLP) over the mechanisms underlying human linguistic capabilities, and the best methods to represent them computationally. Pretrained Language models (PLMs) have even been proposed as proxies that are easier to study than language processing in the human mind, but first, it will be necessary to assess how well they currently model language, and to investigate the mechanisms by which they do it. This thesis proposes to do so with diverse and novel methodology from Linguistics, enabling us to target rarer and less compositional phenomena, which may challenge the models.

To develop methods for **evaluating PLMs' linguistic capabilities**, we first propose to evaluate their ability to represent and learn constructions. Constructions are form-meaning pairings at any level of granularity. A classic example of a well-described construction is the English Comparative Correlative, i.e. *The X-er, the Y-er.* We develop novel probing and evaluation methods, and show that modern PLMs have mostly acquired the syntactic structure of constructions, but even state-of-the-art large PLMs struggle with the non-compositional meaning attached to them. We also evaluate PLM's ability for morphological generalisation, which is the process of applying some learned pattern to the formation of new words. We find that while PLMs are remarkably human-like in their generalisation to novel words, they still make errors and rely on different mechanisms than humans. These results show that while large PLMs have come remarkably close to human linguistic capabilities, we can still find areas where improvement is necessary.

Examining **what modern NLP can contribute to Linguistics**, we first tackle the lack of annotated data for Construction Grammar (CxG). As it is currently not possible to fully automatically annotate or parse constructions, we propose human-in-the-loop strategies to aid linguists in creating corpora. We show the results of a community project to introduce a CxG layer into the Universal Dependencies treebanks. We further develop a hybrid annotation pipeline that uses large LMs to reduce human annotation effort, therefore enabling the cost-efficient creation of corpora for very rare phenomena. Lastly, we show how highly parallel corpora can be used for the unsupervised induction of morphological structure for low-resource languages.

Acknowledgements

I would first like to thank my advisor Hinrich Schütze, for challenging me to be a better researcher, teaching me how to choose interesting projects, and supporting me in developing my own research agenda. I'm also grateful to my thesis committee members, Barbara Plank and Hans-Jörg Schmid, for their advice and support. I would also like to thank Alex Fraser for his support, encouragement, balanced advice, and chocolate, and for having been the first to believe in me as a researcher. My work was supported by the Munich Center for Machine Learning, who funded my conference travel and thereby enabled me to expand my horizons and meet many of the wonderful people who I'm about to thank.

When I first joined CIS, I was encouraged, supported and inspired by Nina Pörner and Timo Schick, for which I remain very grateful. Even more so, I must thank Valentin Hofmann, without whose advice, encouragement and reading lists I would never have been able to articulate what I was interested in, let alone research it. I'm also grateful to the many students I was lucky to work with, chief amongst them Shijia Zhou and Ingo Ziegler, for allowing me to discover how much I enjoy mentoring. The friendly and collaborative work environment at CIS greatly enhanced my enjoyment of my time there. In particular, I would like to thank Viktor Hangya, who patiently listened to my rants over coffee, Masoud Jalili, who was always generous with his food and his time, and Ali Modarressi, who cheered me up with updates from the film industry and his project.

I would also like to thank my collaborators and friends from outside of LMU, who collectively allowed me to feel like a part of an inspiring community of like-minded researchers. I'm grateful to Kyle Mahowald, Kanishka Misra, Adele Goldberg and Reut Tsarfaty, who all gave me energy in the last months of my PhD by making me feel excited about the future. I was lucky to find a community at a Dagstuhl seminar and the related EU COST Action UniDive. In particular, I would like to thank Joakim Nivre, Nathan Schneider, Bill Croft, Nurit Melnik and Agata Savary, who all encouraged me and expanded my research horizons.

Large parts of my work on Construction Grammar were developed during my research stay at Llab at CMU. I first want to thank my fellow students at LTI for making me feel welcome, in particular Maria Ryskina, Naoki Otani, and Brendon Boldt at Llab, as well as Torsten Wörtwein and Leena Mathur at the wider LTI. Most of all, I'm grateful to my two advisors at Llab, Lori Levin and David Mortensen, who helped me develop my research agenda at a crucial point in time, and continued to be influential in my research long after I physically left CMU.

On a more personal note, I am indebted to Lori Levin, for opening her home to me during my time at CMU. She always had an open ear for problems of any kind, and I fondly remember many late-night conversations about linguistics and the world in general. I greatly benefited from continued collaboration through the latter half of my PhD with David Mortensen, whose creative research ideas and helpful advice showed me how fun research can be. His kindness, patience and sense of humour made him a role model in academia and beyond. As an exemplar of the million questions that I must have asked him, I also thank him for entertaining my question about how to write acknowledgements.

I am most grateful to my fellow PhD students turned friends at LMU and beyond, who together made the years preparing this thesis some of the happiest of my life.

Verena Blaschke sadly arrived late into my time at LMU, but made the limited time all the more special by always willing to go on tangents and to the bouldering gym with me. Any good layouting in this work is mostly due to her love for detail. Nina Böbel, who I was lucky to meet at Dagstuhl, I thank for some of the hardest laughs of my PhD, and for enabling me to brush up on my linguistics by never judging any of my ignorant questions. Antonis Maronikolakis, before he moved on to more exciting places, reminded me to enjoy life, not least by being happy to go swimming in the creek behind the office on hot summer days. I'm grateful to him for his empathy and understanding, for

the many mock fights over lunch, and for being a constant living reminder of a world outside of tech. My many travels during my PhD would never have been this fun without Omer Goldman, my companion through many adventures both around the globe and into the world of Linguists. He was the first person who I met at my first conference, which was a borderline miraculous stroke of luck.

Finally, I would like to thank my office mates at CIS, who know very well that they are so much more than that, and that I could not have made it without them. Kerem Senel never gave up on telling me that everything would work out and sometimes succeeded in making me believe him. He sat next to me for over three years and during those shared his snacks, his time, his jokes, and made the office feel more like home. Latif Köksal, one of the most empathetic people I've ever met, I should first thank for being the ideal collaborator through many exciting projects. Even so, I am even more grateful for the many long talks about career and life choices, and for all the jokes mostly communicated by telepathy. Finally, I'm thankful to have shared my time at CIS with Silvia Severini, the kindest and most supportive friend and labmate that I could have hoped for. I thank her for her time, her thoughts, her advice, her pasta and her swimming lessons, and for lighting up the office.

In a very different category, but most of all, I owe thanks to my parents, whose constant support, encouragement, and comfort brought me through the challenging and the exciting times.

Contents

Part I

Introduction

Chapter 1

Introduction and Motivation

> I do believe that Large Language Models should be causing a rethinking of the foundations of Linguistic Theory.
>
> —*Christopher D. Manning*
> *Keynote, Empirical Methods in Natural Language Processing 2023*

This thesis presents new methodology to evaluate the linguistic capabilities of Pre-trained Language Models (PLMs) with regards to their ability to learn constructions and morphologically generalise. We further present ways of contributing to linguistic research and data collection with methods from modern Natural Language Processing (NLP).

We frame this as a contribution to the current debate in both Linguistics and NLP over the suitability of state-of-the-art PLMs as models, or even stand-in experimental subjects, for the widely debated processes through which humans learn, understand, and produce language.

This debate is especially important as it has deep roots in both Linguistics and NLP, which we will now briefly summarise in Section 1.1. We then elaborate on the specific motivation of the work presented here, in the context of this debate, in Section 1.2. To conclude this introduction, we provide detail for our research questions, approach, and contributions, for Construction Grammar (CxG) and Morphology, respectively.

1.1 A Short Joint History of NLP and Linguistics

What is the internal structure of language and how do humans acquire it? How do children who share the same native language all come to acquire the same intuitions of it, and form generalisations and new sentences? How can we describe what they have learned, and how do we apply it to form new sentences and understand given ones?

In the early 1950s, the study of these questions was dominated by the school of American Structuralism, also known as Distributionalism (Bloomfield, 1926; Harris, 1954; Swadesh, 1950; Sapir, 1929; Boas, 1889). They proposed a method for the discovery of the structure of language, which would establish elements and structures based on their usage in context, in a sufficiently large corpus of texts. Even though this method was largely unimplemented due to the technical limitations of the time, the hypothesis of Distributionalism was that it would be possible to learn these structures from a corpus.

Concurrently, as a first implemented computational model of language, Markov chains were utilised by Shannon (1948) as simple n-gram models of human language and quickly gained popularity in language modelling and beyond. This development was famously criticised by Noam Chomsky (Chomsky, 1956, 1957; Miller and Chomsky, 1963), who argued that Markov Models are "surely inadequate for the purposes of grammar", and that the notion of the probability of a sentence is "an entirely useless one" (Chomsky, 1969). The influence of this work was sufficient to halt the further consideration of statistical models in Linguistics for decades. A second key notion of Chomsky's criticism was that of the *poverty of the stimulus* (Chomsky, 1957, 1987), which is his argument that children must be born with some innate knowledge of the structure of language, as they are not exposed to sufficient input to enable them to learn it otherwise.

In this relatively short period of time, two major themes emerge.

The first is the central disagreement between Chomsky and his supporters, and the opposition. Chomsky concluded that language is not learnable unless humans have innate biases that are specific to language, such as a fixed set of principles that all languages follow and a fixed set of parameters in which they vary, all of which would be genetically predisposed (Chomsky and Lasnik, 1993). Opponents, for example from Cognitive Linguistics (Croft and Cruse, 2004; Lakoff, 1987; Langacker, 1986; Ungerer and Schmid, 2006), argue that no such universal grammar (UG, Chomsky, 2006) is necessary, and that we can acquire language solely from data, and our general cognitive biases (Croft and Cruse, 2004).[1] The two sides also disagree about the nature of the data. Chomsky proposes Transformational Grammar (Chomsky, 1957, 1965), the theory that all sentences have a deep structure, which undergoes transformations to reach the surface structure that we can observe. Construction grammar argues (Goldberg, 2006) that “what you see is what you get”, and that deep structure is not cognitively plausible.

The second is the shared aim between computer scientists and mathematicians on the one hand and linguists on the other: **to build a working model of language**. While the motivation for a computer scientist might be application-based, and they might be content to simply improve the model step by step to increase its usefulness, a linguist is motivated by the broader consequences. As even one of the first disagreements in modern linguistics shows: a theory of language may be confirmed by building a model based on it, if it perfectly replicates human behaviour, given the same input as humans. Conversely, a theory of language may be criticised, or even temporarily considered disproved, if critical flaws in its language model are discovered. Such criticism can usually be divided into two categories:

1. at specific and provable points, the does not perfectly replicate human behaviour

2. the was not provided with the same input as an average human

We now briefly summarise the most important points in this debate as it progressed through the development of better language models.

The Markov models first proposed by Shannon (1948) were picked up again for speech recognition in 1975 (Baker, 1990; Jelinek et al., 1975; Baker, 1975; Bahl et al., 1983; Jelinek, 1990), and later used in connection with some of the first works proposing an implementation of distributional semantics for natural language processing (Schütze, 1992, 1998, 1995).

On the linguistics side, the first neural network to be implemented for this debate was that of Rumelhart and McClelland (1986), who built a simple connectionist model for the acquisition of English past-tense formation. Flaws in this model and its imperfect replication of human behaviour were pointed out by Pinker and Prince (1988). They found flaws from both categories that we outlined above: 1) that the output of the model for many words did not correspond to human intuition, and 2) an overreliance on manually encoding various phonological features of the input.

While this influential critique discredited neural networks in the eyes of many linguists, their advancement continued in Natural Language Processing. This led to a perceived contradiction between the linguistics and the development of working models, summarised best in the famous quote “Every time we fire a phonetician/linguist, the performance of our system goes up” (Frederick Jelinek, 1985, as reported by Moore (2005)).

This perceived gap may have increased because, in contrast to the common goal of developing a language model, efforts in both linguistics and natural language processing at the time were very much focused on specific tasks. Advancements were being made using statistical and early neural models for applications such as part-of-speech tagging or machine translation (Brown et al., 1993). At the same time, more linguistics-inspired work in the earliest general meetings of the Association for Computational Linguistics (Sondheimer, 1979, 1980) focused on logical implementations of formal theories of language, such as logical semantics (Barwise, 1981; Palmer, 1981), unification-based grammars (Neumann and Finkler, 1990; Emele and Zajac, 1990) or statistical parsers for hierarchical sentence structures (Martin, 1980), perhaps creating the impression of a dichotomy between purely formal linguistics and purely neural language processing.

NLP steadily progressed with the popularisation of word embeddings, which were demonstrated to be useful for next word prediction (Bengio et al., 2003, 2006), and for word meaning in a variety of tasks (Collobert and Weston, 2007, 2008; Collobert et al., 2011). A major advancement was made by Mikolov et al. (2011) who used recurrent neural networks (RNNs) as language models,

[1] For an introduction to Cognitive Linguistics, see Chapter 2.

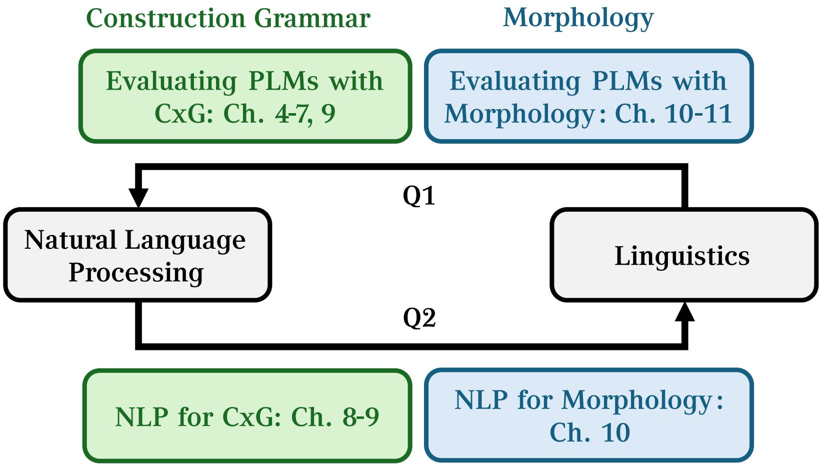

Construction Grammar
Morphology
Evaluating PLMs with CxG: Ch. 4-7, 9
Evaluating PLMs with Morphology: Ch. 10-11
Q1
Natural Language Processing
Linguistics
Q2
NLP for CxG: Ch. 8-9
NLP for Morphology: Ch. 10

Figure 1.1: A visual overview of our research directions and the works included in this thesis. Green indicates Construction Grammar, blue indicates Morphology. The top two boxes are contributing from Linguistics to NLP (**Q1**), while the two bottom boxes contribute from NLP to Linguistics (**Q2**).

and showed that word embeddings could be trained using a language modelling objective (Mikolov et al., 2013a,b).

The success of RNN-based models enabled Kirov and Cotterell (2018) to revisit the debate over English past-tense by training an RNN-based encoder-decoder model.[2] Crucially, they generalise the problem setting to the production of any English verb form from the lemma. They find that their model improves on previous efforts on English past tense, but warn against taking it as a proxy for child language learners, as they do not observe some of the patterns expected during a child's process of acquiring the English past tense.

This is an example of a larger trend that comes out of the advances in language modelling: **both linguistics and engineers can now work on the same language models**. This becomes even more apparent with the introduction of pre-trained language models (PLMs), which are a main focus of this thesis.[3] They are suitable both for adaptation by fine-tuning to any given NLP task, and to linguistic investigation of the base model.

1.2 Motivation

How do language models learn? Interpretability (Belinkov et al., 2020) of models seeks to explain their behaviour and develop ways of investigating their underlying mechanisms and the reasons for their success.[4] Interpretability and evaluation are thus well-suited to help assess the first potential criticism above: has a model learned everything about language that a human has?

PLMs are essentially a black box (Belinkov et al., 2023), which means that any evaluation has to be targeted, meaning that an input will be given to the model, and a specific behaviour observed, such that inferences can be made about the underlying mechanisms of the model.

Necessarily, the evaluation of the linguistic capabilities of PLMs research must come from the perspective of a certain linguistic theory; specific aspects of language must be tested, and the desirable behaviour specified. We claim that this is dangerous if PLMs are eventually to be used as evidence in the debate over human language processing; if the only tool we have is the hammer of testing against rule-based grammar, everything might look like a nail, or rather, like a model that implements syntactic rules. Or to phrase it differently, if we judge PLMs' linguistic competency by their adherence to formal syntactic rules, the community may under- or overestimate their performance, and deny them their place in the debate about the acquisition of language.

As a solution, in this work, we propose to diversify the theoretical standpoints that underlie the linguistic evaluation of PLMs and develop evaluation methods based on areas of Linguistics that have been understudied or -used in NLP.

[2] For more details on this debate, see Section 2.2.1.1

[3] For an introduction to PLMs and an overview of the specific models used in this work, see Section 3.2

[4] For a more detailed introduction, see Section 3.3.2.

Ch.	Construction(s)	Evaluation Contributions	Data Contributions
4	-	Review of prior work	Review of prior work
5	Comparative Correlative	BERT, RoBERTa, DeBERTa	Small annotated corpus
6	Comparative Correlative	BERT, RoBERTa, DeBERTa, OPT	Small annotated corpus
7	Causal Excess, Licensed Causal, Licensed Non-causal	GPT-3.5, GPT-4, Llama 2	Small annotated corpus
8	Interrogative, Existential, Conditional, Resultative, NPN	-	Automatic annotation in UD
9	Caused-Motion	GPT, Gemini, Llama 2, Mistral	Annotation pipeline, small annotated corpus

Table 1.1: Overview of the chapters in Part II

Our work will primarily focus on areas where the community may have overestimated the linguistic capabilities of PLMs, by identifying challenging constructions and showing that PLMs have not fully learned them. However, this can also show areas of underestimation, where formal grammar is simply not a good description of either human or model behaviour. As Baroni (2022) argues:

> Neural language models, by inducing a large set of context-dependent and fuzzy patterns from natural input, and by being inherently able to probabilistically generate and process text, should be better equipped to handle phenomena such as polysemy, the partial productivity of morphological derivation, non-fully-compositional phrase formation and diachronic shift.

Our first research question is therefore:

Q1: How can we diversify the evaluation of linguistic capabilities of PLMs?

Even though it is clear that this is an ongoing field of research and that language models are by no means perfect, they have nevertheless reached a level of performance where they can be used, with guidance, in aiding linguistics research. Our second research question is therefore:

Q2: Given the current state of the art, how can NLP already contribute to Linguistics?

Approach from two perspectives In this work, we focus specifically on Construction Grammar and (word-paradigm) Morphology. Both have so far been under-studied in NLP, and we will argue in the following that both can make important contributions in evaluating PLMs.

For both areas, we make contributions to both research questions. For Construction Grammar, we present our contributions to **Q1** in Section 1.3.1 and to **Q2** in Section 1.3.2. For Morphology, we present our contributions to **Q1** in Section 1.4.1 and to **Q2** in 1.4.2.

1.3 Construction Grammar

Construction Grammar (Goldberg, 1995; Croft, 2001) posits *constructions*, pairings of form and meaning, as the central unit in language (Goldberg, 2003).[5]

We propose to use Construction Grammar to help evaluate how far PLMs have come in emulating human language behaviour. Motivated by the difficulty of procuring corpora for this effort, we further develop methods of creating new corpora for constructions, which we hope will also be helpful for linguists. We present an overview of the chapters dealing with Construction Grammar in Table 1.1, organised by constructions covered, and contributions to both evaluation of LLMs and collection of CxG data.

We first turn to **Q1**, the development of novel evaluation methods for LMs using a more diverse perspective on Linguistics, in this case from CxG. Within CxG theory, most evaluations of LMs could be said to be using CxG, as everything is a construction (Croft, 2001). However, given the current state of LM evaluation, we see the greatest possible contribution from CxG to be the individual constructions described in the literature to illustrate CxG and contrast it with other theories. Such constructions are almost always highly idiosyncratic, have interesting syntactic structure and carry

[5] For an introduction to Construction Grammar, see Section 2.1.

non-compositional meaning (Kay and Michaelis, 2019; Schmid, 2007). We hypothesise that exactly these qualities might make them challenging for LMs, and our approach is therefore to choose constructions from the literature and develop novel methods for evaluating LMs' understanding of them.

1.3.1 How Well Have PLMs Learned Constructions?

The evaluation of the syntactic capabilities of LMs represents an important strand of research in interpretability, in which several strategies have evolved. The two main ones are probing classifiers, where a simple classifier is trained to extract the sought-for linguistic property from a contextual embedding, and behavioural testing, where a test is given to the model that it will only be able to answer if it has learned the property in question.[6] Strikingly, most of the previous work on the linguistic information captured in PLMs has been from a perspective of formal grammar. We hypothesise several reasons for this popularity.

The first is the availability of existing datasets. For example, numerous studies have evaluated PLMs against the Universal Dependencies dataset (Hewitt and Manning, 2019; Manning et al., 2020; Müller-Eberstein et al., 2022)[7]. As a large, coherent, existing and readily available dataset, UD is very accessible, particularly for NLP researchers coming from Computer Science.

The second reason is the simple relationships represented in formal notions of grammar. For example, Hewitt and Manning (2019) test if the token to which the most attention is given from another token is also the one to which it is connected by an edge in the UD dependency tree. This unique methodology is only applicable to a theory in which a word is always connected to exactly one other word.

The third reason is the availability of minimal pairs. Minimal pairs are a device from linguistics literature, where a pair of sentences, one grammatically acceptable and one not, differ only in one specific property, and are therefore used to demonstrate the effect of this property on grammaticality. For example, in the sentence "This thesis [MASK] great", comparing the likelihoods of "is" and "are", we can evaluate specifically if the model is capable of subject-verb number agreement (Wei et al., 2021), as nothing else would influence this choice. This is uniquely suited to the evaluation of PLMs in a controlled setting, as for models trained with masked language modelling (cf. Section 3.2.2), the probabilities of different tokens may be measured in the one position where they differ (in the example, X), or for auto-regressive language models, the perplexity (cf. Section 3.3.1.4) assigned to the whole sentence may be measured for both sentences in the minimal pair, without fear of confounding factors. Minimal pairs are either collected from linguistic textbooks (Warstadt et al., 2020), or can even be automatically created by applying simple rule-based changes to existing sentences (Wei et al., 2021).

We see this as problematic for two reasons.

- **Problem 1** a false dichotomy may be created, as the community, particularly outside of the niche of Computational Linguistics, sees these normative evaluations and datasets as the absolute truth, against which LMs are to be evaluated, and which they must be developed to achieve human language behaviour. For example, Wei et al. (2021) criticise BERT for being overcome by frequency effects instead of perfectly following the rule of subject-verb-agreement.

- **Problem 2** it may lead to an overestimation of the linguistic behaviour of PLMs, as the exact properties outlined above are likely to make the phenomena easier to learn for models. This overestimation may lead to claims of human or even superhuman performance (Dale, 2021; Haider, 2023), which underrepresent the remaining challenges.

1.3.1.1 Approach

We seek to alleviate these problems and approach **Q1** from the perspective of CxG, we develop probing methods for evaluating PLM's knowledge of constructions. In Chapter 4, we argue how Construction Grammar relates to each of the problems outlined above, summarise the little work that has been done in this direction, and propose solutions. First, we argue that CxG-based probing would help with **Problem 1**, because it will introduce more diversity into the conversation about the ideal linguistic behaviour for PLMs. CxG might even be a more natural theory of grammar to

[6] For a more detailed review of these methods, see Section 3.3.2.

[7] For an introduction to UD, see Section 3.1

evaluate LLMs against, as it is usage-based, does not require hierarchical structure, and accounts for frequency effects (Goldberg, 2024). Aside from these theoretical considerations, this will help with **Problem 2**, as literature in CxG mostly focuses on "interesting" constructions, those that are non-compositional in meaning and have surprising syntactic structure.

Prior work on evaluating PLMs on CxG has been limited. Tayyar Madabushi et al. (2020) investigated how well BERT (Devlin et al., 2019) can learn to classify whether two sentences contain instances of the same construction, where the constructions are automatically induced using a modified algorithm from Dunn (2017).[8] Li et al. (2022) focus on argument structure constructions (Goldberg, 1992) and automatically generate instances for a sorting task, in which PLMs, as well as humans, prefer to sort sentences by construction, rather than by verb. Somewhat problematically, the instances are not manually verified and as argument structure constructions have subtle semantic constraints, many of the generated sentences do not hold up upon inspection.[9] Lastly, Tseng et al. (2022) filter a pre-existing list of Chinese constructions (Zhan, 2017) down to those that are easily automatically identified, and find that PLMs seem to have learned about the fixed and the more open slots in the constructions, as they find the more open slots more difficult to predict when masked. While more comprehensive, this evaluation remains fairly superficial.

In developing our own probing studies for CxG, we, therefore, have two goals: a) to push the boundaries of the previously introduced probing methodology to make it more applicable to constructions, and b) to choose constructions from the literature that we hypothesise might be challenging for language models, with the aim of showing their current limitations and inspiring further development to overcome them.

1.3.1.2 Contributions

In Chapter 5, we develop our first probing method for a specific construction, the English Comparative Correlative (CC, Culicover and Jackendoff, 1999), also known as the "the xer, the yer" construction (e.g., "the more, the merrier"). Our main innovation is to consider the syntactic component of the construction as well as the semantics. We adapt the methodology of training a probing classifier with minimal pairs as described above in Section 1.3.1, and automatically generate instances of the CC, as well as structurally similar non-instances, to form minimal pairs. We use this setup to evaluate BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021) at different sizes for their recognition of the structure of the CC. We further design a behavioural task in which the models have to apply their understanding of the CC in context to predict the correct word for a masked token.

As this specific setup was only compatible with PLMs trained on masked language modelling, but the state of the art progressed in the direction of larger PLMs trained on an autoregressive language modelling objective (cf. Section 3.2.3) we extend our methodology in Chapter 6. As smaller PLMs were found to be proficient at the syntactic task but not the semantic one, we specifically focus on this. We adapt our setup such that instead of measuring the probability of the masked token, we instead measure the perplexity of entire sentences to assess if the model is able to use the CC's meaning in context.

Despite interesting results, our evaluation setup has some limitations, which we address in the following chapters. First, the construction of minimal pairs was not organic and even after tweaking, it was still difficult to guarantee that the pairs really were minimal, which we circumvented by using two different setups with different workarounds for this issue. Second, there were no guarantees that models of this size and general capability would be able to understand the setup of the semantics task, so a lack of satisfactory performance might not entirely be due to problems in understanding the construction.

Returning to these issues in the age of Large Language Models (LLMs), we find solutions to both issues in Chapter 7. We solve the problem of minimal pairs by finding a "naturally occurring" minimal pair of two constructions: lexically licensed finite complement clauses, either non-causal ("I was so certain that I saw you") or causal ("I was so happy that I was freed") on the one hand, and the Causal Excess Construction (Kay and Sag, 2012; Fillmore et al., 2012) ("I was so happy that I cried") on the other. The syntactically identical structure accompanied by three different meaning components regarding causality presents an ideal test case for the understanding of these constructions. The second problem was that of ensuring that the behavioural task itself is feasible,

[8] For a more detailed discussion of the problems associated with automatically inducing constructions, see Section 1.3.2.

[9] For a more detailed discussion of argument structure constructions, see Chapter 9.

and that failure can therefore be attributed to a lack of understanding of the construction. We solve this both by evaluating larger LMs (GPT-3.5, GPT-4 (OpenAI, 2022), Llama 2 (Touvron et al., 2023b)) and by testing the understanding using the well-established task of Natural Language Inference (NLI, Bowman et al., 2015).

But how can we evaluate the understanding of constructions when we are not lucky enough to find a trick that uses an established task? One advantage of using instruction-tuned LLMs is that we can now expect them to be able to answer simple questions about their understanding of a sentence and what is happening in it. We use this to our advantage in Chapter 9, where we evaluate LLMs for their understanding of the caused-motion construction (CMC, Goldberg, 1992), e.g. “She sneezed the foam off the cappuccino”. We again form a sort of minimal pair, in making the motion component explicit by replacing “sneeze” with “throw”. About both sentences, we ask the model “In this sentence, is the foam moving?”. The idea behind this evaluation is that a model that lacks understanding of the CMC will answer “no” to the original sentence, but “yes” to the version with an explicit motion verb.

1.3.2 How Can We Use NLP To Help Annotate Data for Construction Grammar?

A recurring theme in the work described above has been the lack of data to perform probing at the scale that is necessary for drawing statistically valid conclusions. In our own work, we have either artificially generated data at a larger scale (Chapters 5, 6), or manually collected annotated data at a much smaller scale (Chapters 5 - 7). Both approaches carry disadvantages: automatically generating data is difficult for semantically more complex phenomena, and even for questions of syntactic acceptability, brings with it the challenge of ensuring that there are no easily exploitable cues for the model, as generated data may not have the full spectrum of variability found in language (cf. Chapter 5). Manually annotating data carries the obvious disadvantage that manual labour needs to be carried out, which limits the size of the dataset and therefore endangers the statistical significance of the result.

There is another dimension to this lack of data: the lack of datasets covering more than one construction, which would make it possible to make broader claims about PLMs such as the ones in Hewitt and Manning (2019).

We argue in Chapter 4 that a possible solution could come from constructicons, which are databases of constructions for one language, sometimes with examples, compiled by linguists.[10] However, in their current state, no constructicon is ready to be used for computational purposes: only some of them are available to freely download, they never include more than a few example sentences, and descriptions of constructions are mostly freeform text and not systematic enough to be used without significant manual work.

Finding a solution to this will contribute to **Q2**, as creating CxG datasets for the purposes of evaluation will also benefit corpus-linguistic research on the same constructions.

1.3.2.1 Approach

We advocate for the use of more natural data in using CxG to evaluate PLMs, and propose larger-scale solutions, which we hope will be more systematic than the current practice of creating a custom solution for every study involving one construction.

Our key ideas are as follows. First, it is vital to rely on pre-existing resources, even if those resources are built with the assumption of conflicting grammatical theories. This can mean either the usage of automated methods such as part-of-speech taggers (Harris, 1965) and dependency parsers (Nivre, 2003) on large-scale corpora, or the search for constructions in pre-annotated treebanks such as Universal Dependencies (UD, de Marneffe et al., 2021). This includes a size-accuracy trade-off: despite high accuracy for state-of-the-art parsers, they may introduce errors, particularly for rare and non-compositional constructions, while annotated treebanks may be too small to include sufficient instances of rare constructions. We further argue that this means no theoretical commitment to any link between CxG and dependency grammar, but rather the usage of corpora and parsers as a means to an end. Second, that manual verification of the produced output is key. As we have already demonstrated in Chapters 5 - 7, and 9, some of the most interesting constructions are impossible to fully automatically annotate with current methods. Our approach therefore

[10] For a more detailed overview of constructicons, see Section 2.1.4.

emphasises the importance of either manually verifiying the output of any automatic process, or critically evaluating it with the help of trained linguists.

Crucially, we see the creation of larger corpora for CxG not only as helpful for further CxG-based evaluation of PLMs (**Q1**), but also as a contribution to the CxG community in Linguistics (**Q2**), where corpus-based studies have been facing the same challenges. We hope that our efforts for corpora creation can enable more quantitative studies of constructions and their distributional properties in the future.

1.3.2.2 Contributions

Our first contribution to this challenge is the product of a community effort, which began at the 2023 Dagstuhl Seminar on Universals of Linguistic Idiosyncrasy in Multilingual Computational Linguistics (Baldwin et al., 2023). We argue for the creation of a construction layer in UD, both for the purpose of creating larger datasets for CxG, and for the enhancement and greater crosslingual comparability of UD. This should serve the purposes of both communities: from a CxG perspective, UD represents a large and well-established community, which we hope can aid in the creation of larger-scale CxG corpora if they are included in the UD annotation schema. From a UD perspective, the annotation of constructions can close gaps in crosslingual comparability, which is one of its major promises.[11] To this end, we present a case study, considering five construction families in ten languages. For each language and construction family, we construct graph pattern queries and match them against UD trees. We then evaluate the challenges encountered, finding that while some constructions were easy to identify, others presented fundamental challenges because of the aforementioned semantic criteria, or were difficult to adequately define across languages. We further propose and implement an annotation schema for UD, in which the heads and constituents of constructions will be marked, and hope that this pioneering study will encourage an effort from the larger community to annotate more constructions.

While this community project will hopefully grow into a larger annotated corpus for CxG over time, it has some drawbacks in its usability for our PLM evaluations, both due to its general approach and its current state of execution. First, as shown in the quantitative evaluation of the project, many constructions are so rare that even the largest treebanks will return fewer than 100 instances, making it unsuitable for work where more instances are required, either to assess the diversity and distributional properties of the instances, or to make statistically sound statements about PLMs' performance on them. Second, there are many constructions with semantic subtleties for which automatic annotation, even with a group of graph pattern queries, will not be possible for sufficient accuracy. These constructions may often be particularly interesting for studying PLMs, as the same semantic subtleties make them more challenging to learn. As a solution to this, we present a hybrid human-LLM corpus construction method in Chapter 9 and demonstrate its usage on the caused-motion construction (CMC, Goldberg, 1992). Starting with the assumption that for such a complex construction, manual annotation is necessary as the last step, we aim to decrease the annotation effort required by filtering the corpus in which the annotator will search for instances, or "concentrating" the instances of the construction. For this, similarly to Chapter 8, we propose the usage of subtree graph queries, but use them on top of a much larger corpus that has been automatically parsed for dependencies. After this prefiltering, we propose to use few-shot classification with GPT (OpenAI, 2022) to further filter the set of potential instances, and finally annotate the result manually to arrive at our final corpus of CMC instances. We give recommendations for prompt design and create a method for calculating the tradeoff between the cost of more elaborate prompting methods and that of human annotation. While we demonstrate this pipeline on the example of the CMC, it can be used for any other construction, which we hope will lead to the construction of more large-scale corpora for rare phenomena.

1.4 Morphology

The morphological capability of humans is the capacity to create words according to systematic patterns of covariation in form and meaning (Haspelmath and Sims, 2010).[12] We consider Morphology to be an important testbed for the capabilities of PLMs, due to its complexity. For example,

[11] For an introduction of Universal Dependencies, see Section 3.1.

[12] For an introduction to Morphology, see Section 2.2.

the English past tense has been documented to show not only irregular forms, but *islands of regularity* (Albright, 2002), where a few words might form the past tense the same way, even though it is drastically different from the regular +ed rule (Bybee and Slobin, 1982). We also expect that PLMs might struggle with morphological tasks due to difficulties with the subword tokenizer, which doesn't necessarily form tokens that correspond to morphemes (Hofmann et al., 2021, 2022).

1.4.1 Have PLMs Acquired a Human-Like Capacity for Morphological Generalisation?

Before the arrival of PLMs, morphological inflection and derivation were studied as NLP tasks. The task of inflection was therefore to generate an inflected form, given a stem and a morphological tag (Cotterell et al., 2017a, 2018; Vylomova et al., 2020; Goldman et al., 2022) and was solved by various trained systems, including encoder-decoder RNN models or transformer models. Similar research focused on modelling derivation (Cotterell et al., 2017b; Vylomova et al., 2017; Deutsch et al., 2018; Hofmann et al., 2020b,c).

With the increasing performance of PLMs, the question of morphological research on LMs shifts. When before, it might have been summarised as "Can we train our model to be able to inflect words correctly?", given the advances of PLMs, it now becomes possible to ask "Have the models acquired morphological capabilities in an unsupervised fashion purely from raw text?". This shift enables us to connect Morphology to **Q1** and make it a further criterion for the human-like linguistic performance of PLMs.

Previous work examining the morphological capabilities of PLMs (Edmiston, 2020; Hofmann et al., 2020a) has focused on smaller models such as BERT. As LLMs have increasingly been claimed to have reached human performance on many linguistic phenomena (Bubeck et al., 2023) claims have been made that they have reached human performance even on morphological generalisation. In early 2023, claims were made on Twitter by leading LLM researchers who had anecdotally tested ChatGPT on the classic example of the wug-test: "This is a wug. Now there is another one. There are two of them. There are two __" and took the model's correct answer as evidence of a groundbreaking level of morphological generalisation. This is problematic, as the original wug-test is now 70 years old and is almost certain to have been seen in the training data of most LLMs, therefore leading to overstated claims of LLM performance. This follows a general trend in the evaluation of LLMs, where training data contamination poses an increasing obstacle to thorough evaluation (Jacovi et al., 2023; Sainz et al., 2023).

1.4.1.1 Approach

Our approach is to study the underlying mechanisms of morphological generalisation in PLMs by using previously unseen data. We adapt the methodology of wug-testing (Berko, 1958), an experimental paradigm in which participants were asked to provide an inflected or derived form of a novel (nonce) word. By creating new nonce words that are guaranteed to not be contained in the training data, we create a rigorous evaluation. Recalling **Q1**, our question is therefore "To which degree are LLMs reaching human behaviour regarding inflectional and derivational morphology?". The advances in LLMs with instruction tuning, which greatly increased their ability to reply to instructions, allow us to almost directly replicate human experimental settings with LLMs, thereby enabling a more direct comparison between human and LLM behaviour. A further component of our approach is to compare the performance of LLMs against that of other systems that have been purpose-built to learn inflection and derivation from training data. There is a long history of these systems with different underlying mechanisms, including rule-learning systems (Liu and Mao, 2016; Wilson and Li, 2021), analogical methods (Calderone et al., 2021), and a transformer-based system (Wu et al., 2021). The usage of these supervised methods not only provides a useful reference point for assessing the performance of LLMs under different conditions, but can also be used to make inferences about the mechanisms underlying LLM performance on morphological generalisation. If we find that LLMs have reached human performance, the next question would be: do they behave more like a rule-based or more like an analogy-based system? One way of assessing this can be to compare each of their predictions on nonce words with that of the LLM.

1.4.1.2 Contributions

In Chapter 10, we introduce the first systematic test of morphological generalisation for ChatGPT. As described above, we create entirely new nonce words in four languages and collect human judgments for them. We then evaluate LLMs on the classic wug-test setup of Berko (1958), and evaluate both their performance and that of state-of-the-art baseline systems against the newly collected human judgments. Crucially, our evaluation takes into account the variability of human judgments: there is no absolute true answer, but rather the human behaviour that should be emulated is given by the frequency distribution over participants' answers. We therefore evaluate the models using top-n-accuracy (cf. Section 3.3.1.3) at different values of n, where n controls how many of the most frequent human answers are considered as correct.

This work's main limitation was its focus on closed-source models, for which we can investigate neither the training data nor the tokeniser or the log probabilities. We were able to assess how far LLMs have come with respect to morphological generalisation, but not which cognitive mechanisms underlie their relative success. To improve on this, in Chapter 11, we take two steps. First, we turn to GPT-J (Wang and Komatsuzaki, 2021), an open-source model trained on an open-source corpus. Second, we choose a phenomenon that is well-suited to the comparison between rule-based and analogy-based models: that of English adjective nominalisation either with *-ity* or *-ness*. We again collect a novel set of nonce words and collect human judgements on them, and compare them against GPT-J prompting results. Crucially, we further compare its results directly against a rule-based and an analogy-based system, to evaluate which provides the better fit for its behaviour. Additionally, we analyse the training data to assess how well GPT-J's predictions match the statistics in the data. As it could be argued that our analysis of GPT-J is specific to the model and would not hold for larger models like GPT-4, we additionally compare against it.

1.4.2 How Can We Leverage Parallel Data for Unsupervised Morphology?

The unsupervised induction of morphological structure has traditionally been a central field of interest in NLP (Yarowsky and Wicentowski, 2000; Goldsmith, 2001; Schone and Jurafsky, 2001; Creutz and Lagus, 2002; Hammarström and Borin, 2011). While a working model for this induction would doubtless be useful for linguistics, especially typology, it is also relevant to **Q2**: if we build a working model, we will have shown that this structure is, in principle, learnable from data.

1.4.2.1 Approach

Our approach to this problem is from a typological perspective, using a highly parallel corpus. The key idea is that in utilising this corpus, we have access to crucial information gained from seeing so many ways to express one sentence, which we can use to induce linguistic structure in an otherwise unsupervised manner. For example, if we have many translations of the same noun phrase available, which will be grouped into different cases by different case systems, the combination of these cases will tell us something very specific about the noun phrase.

1.4.2.2 Contributions

In Chapter 12, we introduce a new task in computational morphology, that of unsupervised case marker extraction. To build a first model for this task, we leverage the Bible corpus (Mayer and Cysouw, 2014), a massively multilingual sentence-aligned corpus, to extract case markers in 83 languages. Compared to other chapters, our approach relies on simple technology, as we require only the corpus, a noun phrase chunker which will extract all noun phrases from each sentence, and a word alignment system (Jalili Sabet et al., 2020).

1.5 Outline

The remainder of this work is structured as follows. Part II contains our work on Construction Grammar, with Chapters 5 through 7 about CxG as a method for evaluating PLMs and Chapters 8 and 9 about NLP-aided data collection for CxG. Part III contains our work on Morphology, with Chapters 10 and 11 about evaluating LLMs for their morphological generalisation and Chapter 12 about the automatic extraction of case markers. The rest of this part provides background information relevant to the publications, which we see as particularly important, given the interdisciplinary nature of this work. In Chapter 2, we discuss the linguistic background, which is divided into Construction Grammar in Section 2.1 and Morphology in Section 2.2. In Chapter 3, we introduce the background in Natural Language Processing, starting with a discussion of Universal Dependencies in Section 3.1 and moving on to an overview of the utilised Pre-trained Language Models in Section 3.2 and an overview of evaluation paradigms in Section 3.3. We conclude in Part IV, which contains a summary of our findings and an outlook into future research directions.

Chapter 2

Linguistic Background

In this section, we introduce the background from the field of Linguistics for this thesis. We aim to thereby make the following chapters more accessible to readers who come from a purely computational background. We first give an overview of Construction Grammar in Section 2.1, and then of Morphology in Section 2.2.

2.1 Construction Grammar

We give an overview of the central themes and ideas of Construction Grammar (CxG), with special emphasis on the theories used implicitly or explicitly in the practical work presented in Sections 2.1.2 and 2.1.3. We also discuss in detail the various efforts to create comprehensive databases for Construction Grammar, called constructions, in Section 2.1.4. For a more general overview of the field, we refer the reader to Hoffmann and Trousdale (2013) and Fried and Nikiforidou (2025).

2.1.1 A Short History of Construction Grammar

Construction Grammar is a theory which posits form-meaning pairings called constructions as the central units of language. These units could be anything from a word or even subword unit to a syntactic construction encompassing an entire sentence, therefore removing the traditional barrier between Lexicon and Grammar (Chomsky, 1965, 1995; Pinker, 1999).

This theory was first developed in opposition to Chomsky's principles-and-parameters approach (Chomsky, 1995) which hypothesises a strict separation between syntax and semantics, and sees constructions such as "The Xer, the Yer" (e.g., "The more I read, the less I know") as peripheral phenomena, or exceptions to the rule. The Chomskian view is summarised by Fillmore et al. (1988) as follows. Speakers have knowledge of the lexicon of their language, including the meaning of words and where they can appear. In addition to this, they have knowledge of the (basic) grammatical rules of their language, which enable them to combine these words into more complex structures. Separately from these, they also know basic semantic interpretation principles, with which they can infer the meaning of entire sentences. To use the sentences in context, they further use their pragmatic knowledge to associate them with particular types of situations.

Crucially, this view of language is entirely compositional: the syntactic acceptability of a sentence is determined by applying the syntactic rules to the lexicon, and the rules of semantic interpretation similarly build the sentence meaning.

Many phenomena in language are incompatible with this idea, as was first pointed out by Fillmore in the pioneering 1985 *Syntactic Intrusions and The Notion of Grammatical Construction* (Fillmore, 1985) who refers to "constructions often endowed with properties which are not independently determined by facts about their constituency or their derivation". Along with constructions, he introduces another notion central to CxG: "single-level representation of complex syntactic objects, as opposed to multi-level or derivational representations". He describes two constructions which he initially calls syntactic intrusions, the redundant *have* in past counterfactual clauses under certain conditions, and the intrusion of phrases like *the heck* into a question. Fillmore argues that the only way to account for these syntactic intrusions that have very specific conditions attached to them is to see them as **constructions**, lexical entries "capable of occupying particular higher-phrase positions in sentences" and including "both the needed semantic role and the needed specification of structural requirements". He also introduced the central disagreement between Construction and Transformational Grammar that would become even clearer later: while neither dispute that constructions exist, the difference is that Transformational Grammar sees them as the *periphery* of language and not its *core*, while Construction Grammar claims that there is no major discontinuity between the two.[1]

It is important to note here that our empirical work does not necessarily take a strong stance on this, as all Chapters dealing with constructions and NLP in a practical way could equally be carried out from a standpoint of Transformational Grammar. The position paper in Chapter 4 discusses the implications of adopting a CxG *constructions all the way down* approach for NLP. For a more detailed discussion of computational methods that explicitly target this debate over rules vs. periphery, see the Future Work proposed in Section 13.4.

To continue the history of CxG, several case studies on specific English constructions followed, arguing that each are incompatible with the Chomskian view of language.

The English Comparative Correlative, also called the "the X-er, the Y-er" construction, was introduced in Fillmore (1986).[2] Lakoff (1987) argues that the semantic difference of the two types of *there*-constructions, deictic ("There's Harry with his red hat on") and existential ("There was a man shot last night") cannot be adequately be accounted for using previous approaches. He further asserts that "grammatical constructions have a real cognitive status" and that a continuum exists between the grammar and the lexicon. He also empathises the link between CxG and prototypes (Rosch, 1973), stating that "prototype-based categorization occurs in grammar".[3] Fillmore et al. (1988) similarly discuss the "let alone" construction, as exemplified by "I barely got up in time to EAT LUNCH, let alone COOK BREAKFAST." He argues that "those linguistic processes that are thought of as irregular cannot be accounted for by constructing lists of exceptions: the realm of idiomaticity includes a great deal that is productive, highly structured, and worthy of serious grammatical investigation". He also introduces the central notion that meaning can be attached to arbitrarily large structures.

These were summarised by Fillmore (1988), who also first proposes to "treat grammatical constructions as syntactic patterns which can fit into each other, impose conditions on each other, and inherit properties from each other", a notion that would later develop into the concept of a construction (Section 2.1.4). He also describes the notion of slots in constructions, which require fillers with certain properties, which will become important in computational work.

Another classic example, the "What's X doing Y?" construction, was introduced by Kay (1999). With the example of "What's this fly doing in my soup?", where the question is not about the fly's activities in the soup, but about the reason for its presence there, to argue for "a grammar in which the particular and the general are knit together seamlessly".

The foundations laid by these classic works have been and continue to be developed. Today, some central tenets are the shared basis for most approaches to CxG (Goldberg, 2013; Ziem and Lasch, 2013). First, constructions show the continuum between lexicon and grammar (Boas, 2010; Broccias, 2012; Goldberg, 1995; Fillmore, 1989), on which all constructions are situated. While the two ends of the spectrum exist, with one-word lexical constructions on the one hand and highly syntactic constructions like the ditransitive on the other, many constructions are idiomatic and only partially schematic, and therefore showcase that the separation between these two extremes can not be cognitively plausible. Second, constructions form a network of hierarchies called a constructicon (Jurafsky, 1992; Boas, 2011; Goldberg, 1995; Fillmore et al., 2012).[4] Third, constructions are prototypical structures that become entrenched with increasing frequency (Goldberg, 2006; Brooks and Tomasello, 1999; Boyd and Goldberg, 2011; Schmid, 2020). Goldberg (2006) requires that constructions be either entrenched in the language community, or contain a syntactic or semantic component that is not predictable by the application of general rules to its components.

Beyond these central tenets, a variety of constructionist approaches have been developed in slightly different directions. We now turn to describing the two which are relevant to this thesis in detail: Goldbergian (Goldberg, 1995, 2006), and Radical Construction Grammar (Croft, 2001). We then briefly summarise other approaches.

[1] Note that this central question of rules and exceptions on the one hand, and a continuum on the other, will come up again in the Morphology part of this work (Part III).

[2] We investigate different PLMs' understanding of this construction in Chapter 5.

[3] This links the CxG part of our work to the morphological part, in which analogy- and prototype-based theories are central.

[4] See Section 2.1.4.

Construction	Examples
Word	Iran, another, banana
Word (partially filled)	pre-N, V-ing
Idiom (filled)	give the Devil his due
Idiom (partially filled)	jog <someone's> memory
Idiom (minimally filled)	The Xer the Yer
Ditransitive construction: Subj V Obj1 Obj2 (unfilled)	He baked her a muffin
Passive: Subj aux Vpp (PPby) (unfilled)	The armadillo was hit by a car

Table 2.1: Constructions at varying levels of complexity and abstraction, adapted from Goldberg (2013).

2.1.2 Goldbergian Construction Grammar

Under the supervision of George Lakoff, Goldberg (1992, 1995) discusses four Argument Structure Constructions (ASCs): i) the ditransitive construction ("Elwin faxed her the news"), ii) the caused-motion construction ("Sam sneezed the napkin off the table"), iii) the resultative construction ("She kissed him unconscious") and iv) the X's way construction ("She smiled her way through the crowd"). Each of these constructions affects both the argument structure of the verb and the meaning of the sentence. She argues that these constructions make it apparent that meaning must attach to the ASC and not the verb, as it would be unreasonable to expect every verb that could ever appear in, e.g., a caused-motion construction, to include this possibility, and the resulting changes to meaning and argument structure, in its entry in the mental lexicon.

Much like Lakoff's earlier work, this approach to CxG focuses on the cognitive plausibility of CxG, and on the mechanisms by which speakers acquire constructions. Goldberg (2013) summarises the main tenets of CxG as follows: i) grammatical constructions are learned pairings of form and function ii) semantics is associated directly with surface form iii) constructions form a network in which nodes are related by inheritance links iv) languages vary in wide-ranging ways.

She includes a fifth tenet which is central to Goldbergian CxG: that it is usage-based and that "knowledge of languages includes both items and generalisations, at varying levels of specificity". A modified version of the examples of constructions at varying levels of complexity and abstraction can be seen in Table 2.1.

Goldberg (2006) elaborates on the theme of acquiring generalisations from input to learn constructions. She presents experiments in which children are shown to acquire novel constructions quickly based on input. Crucially, these experiments are successful without the need for explicit negative feedback about overgeneralisation, which has long been a central claim of Transformational Grammar (Chomsky, 1957, 1965).

Goldbergian Construction Grammar is an implicit basis for Chapter 4, in which we argue that CxG is a good fit to evaluate LLMs, and the main inspiration for Chapter 9, which concerns the caused-motion construction.

2.1.3 Radical Construction Grammar

Croft (2001) proposes Radical Construction Grammar, a theory of Syntax that argues that "virtually all aspects of the formal representation of grammatical structure are language-particular". It suggests using an inductive method of analysis to find generalisations between languages, but simultaneously respect the diversity and the inherent arbitrariness of the world's languages. Croft finds that constructions rarely generalise across languages, and even when they do, they display "wildly different distributions". Perhaps most strikingly, he argues that even parts of speech such as *noun* and *adjective* do not hold up to typological scrutiny without compromises that should not be made.

Most recently, Croft (2022) attempts to describe the "morphosyntax of the world's languages" and therefore takes a view of CxG that is chiefly motivated by typology. A central notion is that a construction not only has a meaning, but also a way of conveying that meaning: information packaging. Croft introduces a functional framework for categorising the functions expressed by language, both in terms of semantic content and in terms of information packaging. This packaging is divided into three categories: reference (what the speaker is talking about), predication (what they are asserting about the referent), and modification (additional information about the referent).

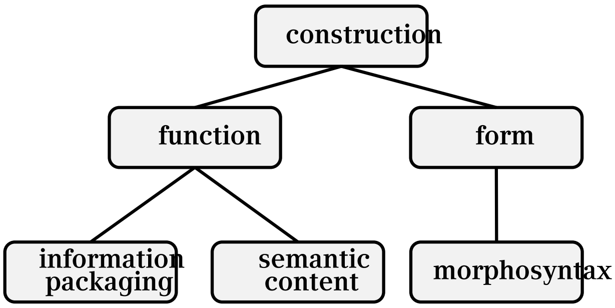


Figure 2.1: The relationship between constructions, information packaging, semantic content, and morphosyntax, adapted from Croft (2022).

For example, a meaning related to "sharp" may be packaged using reference ("the sharpness") or modification ("the sharp thorns"). For Croft, constructions are "any pairing of form and function in a language used to express a particular combination of semantic content and information packaging'. This means that a construction does not define how the function is expressed, but merely what function is expressed. The specific way of expressing a particular function is called a *strategy*, e.g., both English and Spanish can be said to employ a verbal copula strategy for their predicate nominal constructions. The aim for this view of CxG is then to chart all constructions, along with the strategies that exist across languages, and which languages employ which strategy.

Radical Construction Grammar is the theoretical basis for Chapter 8, in which we automatically annotate four constructions and one strategy as defined by Croft.

2.1.4 Constructicons

Construction Grammar in general suffers from a lack of annotated corpora. This is partially inherent to a theory which proposes that a large number of constructions are stored in memory, and multiple constructions are applied in the same sentence and interact in a way that has not been fully specified. The former makes it difficult to build comprehensive lists of constructions (constructicons), while the latter makes it virtually impossible to create corpora in which sentences have been fully annotated with every construction they contain. This stands in contrast to simpler theories of grammar such as dependency grammar, where both exist (cf. Subsection 3.1).

However, several recent projects are attempting to build constructions for their language. An overview is provided by Lyngfelt et al. (2018). We summarise them in Table 2.2. Most efforts are fairly recent and not all are available freely online, and only one, the Russian Constructicon, contains more than 500 entries. While the efforts are currently disconnected, a Constructicon Alignment Workshop was held in December of 2022 at the University of Gothenburg, with the aim of fostering collaboration and a shared data structure.[5] This shared structure is based on the theoretical basis of Croft (2022), who proposes a framework for the constructions of all languages.

The relatively small size of constructions motivates our work on using NLP to annotate more data in Chapters 8 and 9.

2.2 Morphology

Morphology is the study of the internal structure of words (Haspelmath and Sims, 2010). A distinction is typically made between inflectional morphology, which is concerned with different forms of one word (e.g. *draw, draws, drew*), and derivational morphology, which is concerned with families of related words (e.g., *read, readable, reader*). There are three distinct approaches to morphology: morpheme-based, lexeme-based, and word-based. Morpheme-based morphology, also called the item-and-arrangement approach (Hockett, 1947, 1954) analyses words as arrangements of morphemes, which are the minimal meaningful units of language. Lexeme-based morphology (Anderson, 1992), also called the item-and-process approach, sees a word form as the result of the application of some alteration rule to another word form. In contrast, word-based morphology (Matthews, 1991), also called the word-and-paradigm approach, sees morphology not as the study

[5] https://www.globalframenet.org/caw2022

Name	Language	Years	Avail.	Size
CASA (Herbst and Hoffmann, 2018)	English	2018–	LIC	168
German Framenet-Constructicon (Ziem et al., 2019)	German	2017–	Free	244
SweCcn (Lyngfelt et al., 2018)	Swedish	2012–	Free	409
FrameNet Brasil Constructicon (Laviola et al., 2017)	Brazilian Portuguese	2010–	Free	220
Russian Constructicon (Janda et al., 2018)	Russian	2016–2021	Free	2277
(Ohara et al., 2004)	Japanese	2014–	No	Unknown

Table 2.2: Overview of Constructicon Initiatives in different languages, by years active, availability and size (number of constructions). For availability (Avail.), *free* stands for freely available, *LIC* for available, but with some licence restrictions, and *no* stands for not available online.

of morphemes, but rather as "the branch of linguistics which is concerned with the forms of words in different uses and constructions".

A central dichotomoty exists between item-and-arrangement and item-and-processes approaches on the one hand, and word-and-paradigm approaches on the other. While the former assume that morphology, just like Syntax (cf. Section 2.1) is governed by compositional rules operating over subword items (i.e., morphemes), the latter puts forward that words, rather than morphemes are the basic unit of morphology, and morphological behaviour is governed by analogical processes between them.

We now briefly summarise the debate of rules vs. analogy in morphology, which is highly related to Chapters 10 and 11. For a more general introduction, we refer the reader to Haspelmath and Sims (2010).

2.2.1 Computational Modelling of Rules and Analogy

Chapters 10 and 11 are specifically written in the context of the morphological debate about rules vs. analogy, which asks: can the morphological behaviour of humans be learned by a connectionist model (Rumelhart and McClelland, 1986) such as a neural network, or is a rule-based learner required? This is related to larger debates in linguistics, where one side is represented by Chomsky's ideas (Chomsky, 1956; Chomsky and Lasnik, 1993; Chomsky, 1995) that language is a system of vocabulary items and rules that operate on them, and the other side claims that language can be acquired without explicit rules simply on the basis of forming analogies and abstractions over seen input (Croft, 2001; Goldberg, 1995)

2.2.1.1 English Past Tense

The most prominent example discussed in this debate has been the phonological realisation of the English past tense inflection. Most verbs inflect for past tense with either /-d/, /-ɪd/ or /-t/ but some form rare and irregular inflection classes (e.g., "teach" → "taught").

A first computational model of English past tense was presented from the connectionist point of view by Rumelhart and McClelland (1986), who built a simple neural network that they claimed was able to learn English past tense from data without inductive biases or explicit mechanisms for rules. This resulted in a debate that continues to the present day, an overview of which can be seen in Figure 2.2.

Pinker and Prince (1988) proposed a dual-route system with separate mechanisms for rules and exceptions, and pointed out several flaws in the experimental setup of Rumelhart and McClelland (1986).

To test the descriptive adequacy of the models, the main methodology has become the testing of a system on novel words. Following the wug test (Berko, 1958), where children were prompted

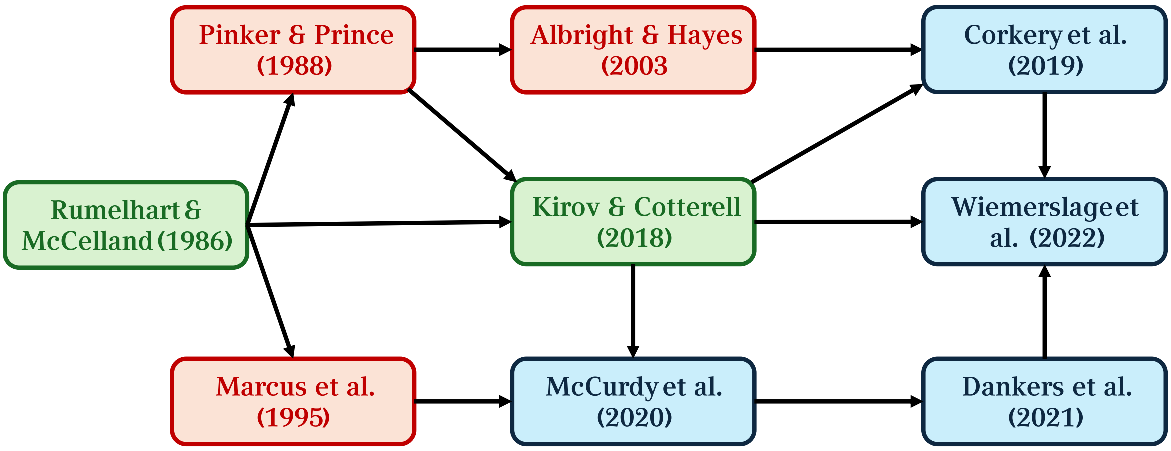


Figure 2.2: Summary of the computational debate over the English past tense and German noun plurals, adapted from Wiemerslage et al. (2023). Green is used for papers finding in favour of connectionist models, while red is used for papers finding against. Blue papers present inconclusive results.

to form, amongst other forms, the past tenses of made-up English verbs such as "wug", several datasets of human past tenses formed for novel verbs have been created (Marcus et al., 1995; Albright and Hayes, 2003). This was first applied to the past tense debate by Prasada and Pinker (1993), who compare the Rumelhart and McClelland (1986) model to human behaviour on the same set of wug words, and find that it cannot account for human generalisation. Albright and Hayes (2003) build an analogy- and a rule-based model and test it on their wug datasets and the frequencies assigned by humans to different inflected forms, and find that the rule-based model matches human nonce word inflection better.

The debate was picked up again by Kirov and Cotterell (2018), who train a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with attention (Bahdanau et al., 2016) and conclude that it solves many of the issues that Pinker and Prince (1988) first observed with the Rumelhart and McClelland (1986) model. This work was again criticised by Corkery et al. (2019), who re-implement the model and perform multiple runs to simulate multiple participants. They find that while the LSTM model outperforms earlier neural network models, it still falls short of rule-based methods, e.g. by overproducing irregular forms.

2.2.1.2 German Noun Plurals

A similar debate with similar methodology has been carried out over the issue of German noun plurals. This is interesting because of the five plural suffixes /-(e)n/, /-e/, /-er/, /-s/, and /∅/, none have a regular majority. Additionally, the inflection class that is most frequently used to generalise to nonce forms, /-s/, is not the most frequent in the regular German data.

An LSTM model for this was presented by McCurdy et al. (2020), who find that the model overgeneralises to the most frequent inflection class in the training data (/-e/), and that the model did not correlate well with human production probabilities in general. Dankers et al. (2021) perform behavioural and structural analysis, and find that RNNs mostly generalise like humans, but also rely on shortcuts, such as word length.

Wiemerslage et al. (2022) additionally train simple transformer models on both English past tense and German noun plurals, and find that while not perfect, they correlate better with human performance than previous models. The debate is therefore still ongoing.

We contribute to these two debates specifically in Chapter 10, in choosing the English past tense and German noun plurals as test cases for the evaluation of morphological inflection in ChatGPT.

2.3 Summary

We have summarised the linguistic background of our work. We have introduced Construction Grammar, placing special emphasis on the approach of Goldberg and Croft, which are used in this work. We have also given an overview of the available constructicons in different languages. We have further given a short introduction to Morphology and elaborated on the computational modelling of rules and analogy. We now turn to summarising the technical background of our work.

Chapter 3

Natural Language Processing Background

In this section, we introduce the background from the field of Natural Language Processing that is required for this thesis. We aim to thereby make the following chapters more accessible to readers who come from a purely linguistic background. The section is divided into two parts: in Subsection 3.1, we introduce the framework of Universal Dependencies, and in Subsection 3.2, we introduce the pre-trained language models used in later chapters.

3.1 Universal Dependencies

As the most successful and widespread linguistic annotation framework in current NLP, Universal Dependencies (UD, de Marneffe et al., 2021) is an important tool for several of the works presented here. It is both a framework for the annotation of parts of speech, morphological features, and syntactic dependencies, and a community-built treebank covering over 100 languages. In the following, we present a brief history and overview of UD, and then describe how it is useful for the computational study of Construction Grammar.

3.1.1 History of Universal Dependencies

Universal Dependencies grew out of several concurrent efforts to create universal schemata for dependency annotation. Stanford Dependencies were originally developed in 2005 for the Stanford parser (de Marneffe et al., 2006; de Marneffe and Manning, 2008). A separate effort, the Google universal tag set, was released in 2012 (Petrov et al., 2012), and used in HamleDT (Rosa et al., 2014), a project that brought treebanks of many languages under a common annotation scheme.

The Universal Dependency Treebank project (UD, Nivre et al., 2016) released treebanks for 6, and shortly after, for 11 languages, and brought together Stanford Dependencies and the Google universal tagset, while the second version of HamleDT (Rosa et al., 2014) provided this annotation for 30 languages. Shortly after, the Universal Stanford Dependencies (USD, de Marneffe et al., 2014 were released, merging the initiatives. Since 2017, a Universal Dependencies Workshop has been held every year, and UD v2 was released in 2020 (Nivre et al., 2020) with updates to the annotation guidelines.

3.1.2 Annotation Principles of Universal Dependencies

The annotation principles of UD have grown from the motivating principles of usefulness for NLP and comparability across languages. This makes it different from other variants of dependency grammar, as it emphasises simple surface representations to allow parallelism between similar constructions across languages, despite differences in word order, morphology, or function words.

An example of this is shown in Figure 3.1, for parallel sentences from English, Bulgarian, Czech, and Swedish. The main grammatical relations involving a passive verb, a nominal subject, and an oblique agent are the same, but the specific grammatical realisation varies.

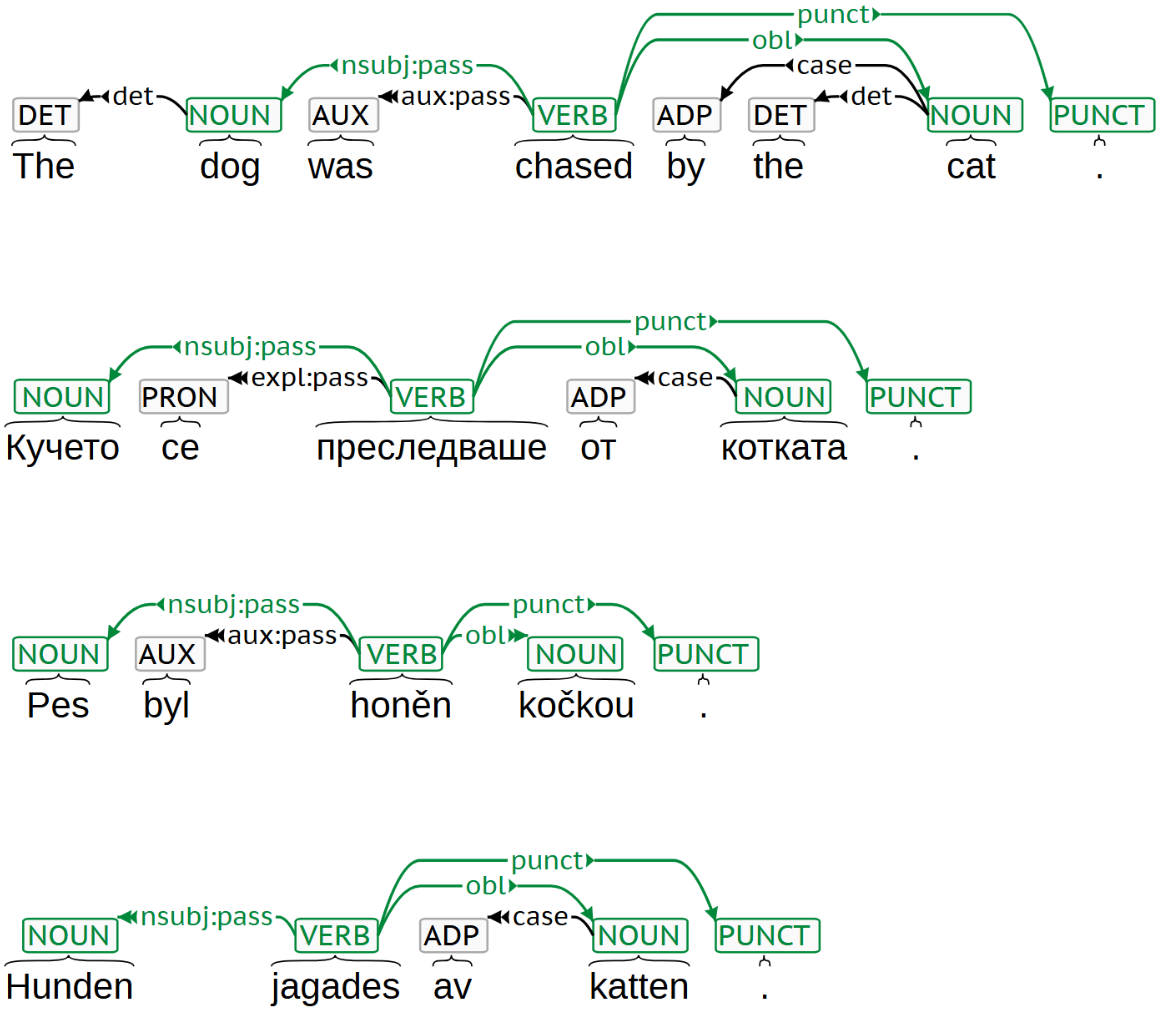


Figure 3.1: Example of the parallel syntactic-semantic structure of UD for the sentence “The dog was chased by the cat” in English, Bulgarian, Czech, and Swedish. Adapted from the Universal Dependencies website `https://universaldependencies.org/introduction.html`.

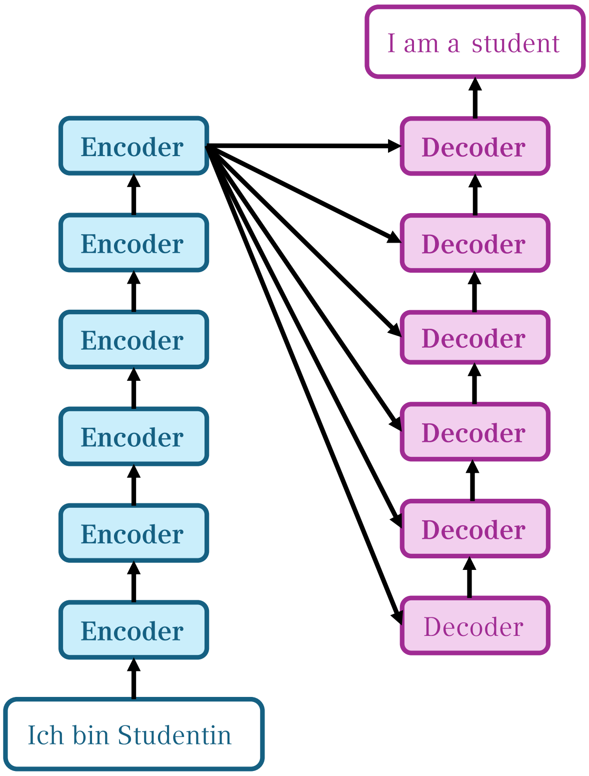


Figure 3.2: An encoder-decoder model for machine translation, translating “Ich bin Studentin” (German) into “I am a student” (English).

3.2 Pre-trained Language Models

The goal of a language model (LM) is to model the properties of language and to be able to generate text. Training a model to generate the next token given some previous input is called autoregressive language modelling. Formally, they factorise the probability $p(w)$ of a text sequence $w = W_i, ..., w_n \in V^*$ as

$$p(w) = \prod_{i=1}^{n} p(w_1 | w_1, ..., w_{i-1}) \tag{3.1}$$

where V is a vocabulary of tokens.

While the first language models were introduced in 1948 (Shannon, 1948), significant advances were not made until the usage of Recurrent neural networks (RNNs, Elman, 1990) was popularised in NLP (Sutskever et al., 2014). RNNs were used to build encoder-decoder models. In an encoder-decoder model, the encoder builds a representation of the input that consists of some kind of hidden state, while the decoder uses this representation to generate text. For example, in a translation task, the meaning of the source sentence is encoded, and then decoded into the target sentence.

A central issue in the development is that of scaling: as LMs have generally shown improvement when trained on more data, how can we build architectures that can be trained so efficiently that it becomes feasible to train them on large amounts of data?

While scaling was difficult for RNNs due to their sequential nature, a significant advancement was made by the introduction of the Transformer.

3.2.1 Transformer

The Transformer (Vaswani et al., 2017) is an encoder-decoder architecture based on attention mechanisms. It was originally created for the task of machine translation (MT).

While neither encoder-decoder models nor attention were a novel idea for MT (Bahdanau et al., 2016), the key innovation was to make both the encoder and decoder rely entirely on self-attention for the computation of the input and output. This enabled it to process inputs in parallel and made it more efficient, contributing to its dominance as a Language Model architecture.

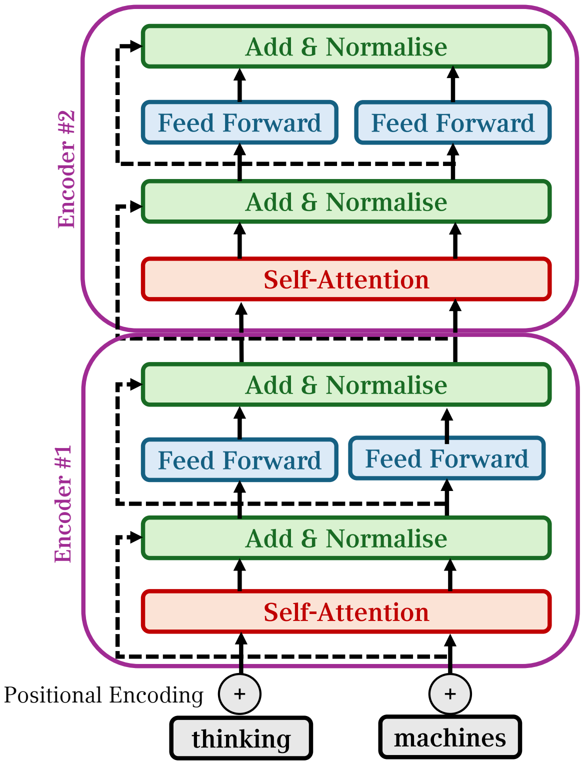


Figure 3.3: Schematic overview of stacked encoder blocks in an encoder-only transformer model.

The basic architecture is shown in Figure 3.2, consisting of an encoder and a decoder. A schematic representation of the encoder and decoder layers is shown in Figure 3.2, for the example of translating a French sentence to English. The encoder consists of stacked encoder blocks, which all have the same internal structure, but do not share parameters. The output from one block are fed as inputs to the next. Each encoder block consists of a multi-head attention and a feedforward layer, with layer normalisation and skip connections in between.

The decoder uses masked multihead-attention during training to ensure that the model does not have access to the correct next word at any given time. In each decoder block, between the masked multi-head attention and the feedforward layer, unconstrained multi-head attention attends to the output of the encoder while using the queries from the previous decoder layer, thus enabling the model to incorporate both the encoded source-language sentence and the partial target-language sentence it has output thus far. An overview is shown in Figure 3.3.

We now describe the main components separately.

Positional Encodings Since the Transformer processes all input tokens in parallel, it has no knowledge of the position of each token in the sentence. This is therefore manually injected via fixed *Position Encodings* (Gehring et al., 2017). Because the position encodings vary slightly for each position, the model can learn different behaviour for the same token in different positions. The encodings are computed as follows:

$$\begin{aligned}\mathbf{PE}_{(pos,2i)} &= sin(\frac{pos}{10000^{2i/d_{model}}})\\ \mathbf{PE}_{(pos,2i+1)} &= cos(\frac{pos}{10000^{2i/d_{model}}})\end{aligned} \tag{3.2}$$

where $\mathbf{PE} \in \mathbb{R}^{T \times d_{model}}$, T is the maximum sequence length, d_{model} is the dimensionality of the embedding vectors, pos is the position in the sequence, and i is the index of the hidden dimension.

Attention Attention is a mechanism by which weights can be learned from each token to each token. This means that the model can learn how important tokens are for each other. In self-attention, the output representation of each token is a transformation of itself and all other tokens' input representations weighted by attention, such that information about the surrounding words can be incorporated and make the representation contextual. This is the central mechanism behind the Transformer architecture.

For each token, the mechanism takes the encoder input and produces three vectors **q**, **k**, and **v**. Given an input sequence $s = (s_1, s_2, ..., s_n)$ with n tokens and their embeddings $\mathbf{X} \in \mathbb{R}^{N \times d_{model}}$, it computes:

$$\mathbf{K} = \mathbf{X}\mathbf{W}^K, \mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \mathbf{V} = \mathbf{X}\mathbf{W}^V \tag{3.3}$$

where $\mathbf{W}^K$, $\mathbf{W}^Q$, $\mathbf{W}^V$ are weight matrices. The self-attention of **X** is then computed as:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{3.4}$$

where the scaling factor d_k is the dimension of the keys. The result is a weighted sum, where the weights are given by a probability distribution determining the attention that should be given to each input token.

To increase the model's representational capacity, Vaswani et al. (2017) propose *Multi-Head Attention*, which consists of m linear projections (heads) for **q**, **k**, and **v** to d_q, d_k, and d_v dimensions:

$$MultiHeadAttention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = concat(head_1, ..., head_m)\mathbf{W}^O \tag{3.5}$$

where $\mathbf{W}^O \in \mathbb{R}^{md_v \times d_{model}}$ is a projection matrix. Given $\mathbf{W}_i^Q$ and $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_k}$, and $\mathbf{W}_i^V \in \mathbb{R}^{d_{model} \times d_v}$, each attention head is computed as:

$$head_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{3.6}$$

The formulation for the attention used in the decoder model is slightly different, as the attention is limited to the previous word embeddings with respect to the current position:

$$MaskedAttention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T + \mathbf{M}}{\sqrt{d_k}})\mathbf{V} \tag{3.7}$$

where the mask **M** is a matrix of zeros and $-\infty$.

The transformer was trained using supervised data: for example, for translation, each source sentence is paired with a target sentence that the model aims to generate. It is therefore clear what the output should be, and the trained model is specific to this task. By contrast, the task of language modelling as introduced above is self-supervised: there is no clear target, rather, the aim of the learning process is to capture the underlying structures of the training data (i.e., a corpus). This is inherently more challenging but also brings greater opportunities: a self-supervised language model will be more versatile and could be adapted to handle different tasks.

Seeking to build a language model based on the transformer architecture, but with only a decoder, which would thus be suited for self-supervised learning of text generation, Radford et al. (2018) present the Generative Pre-trained Transformer (GPT), a 117M parameter model, with limited success. Taking a different approach, Devlin et al. (2019) propose to train an encoder-only model using masked language modelling (MLM). The training idea is to pass the input sequence to the model but choose a subset of positions to be masked. The model is trained on the task of reconstructing these tokens from the surrounding context, thereby forcing the contextualised representations to be useful. For example, in Figure 3.4, in the sentence "how are you doing today?", "you" has been replaced with a [MASK] token, and the model is predicting the word that has been replaced.

With the comparatively limited resources available at the time, this made more efficient use of both training data and computational resources and led to rich contextual representations of tokens. They also introduced the notion of pre-training (training the general model purely on MLM), and fine-tuning (using supervised data and continuing to train the model, so that the knowledge acquired in pre-training can enable better performance on some down-stream task).

3.2.2 Early PLMs: Masked Language Modelling

This idea was used in several pre-trained language models. Below, we briefly summarise those that are used in this work, and provide an overview in Table 3.1.

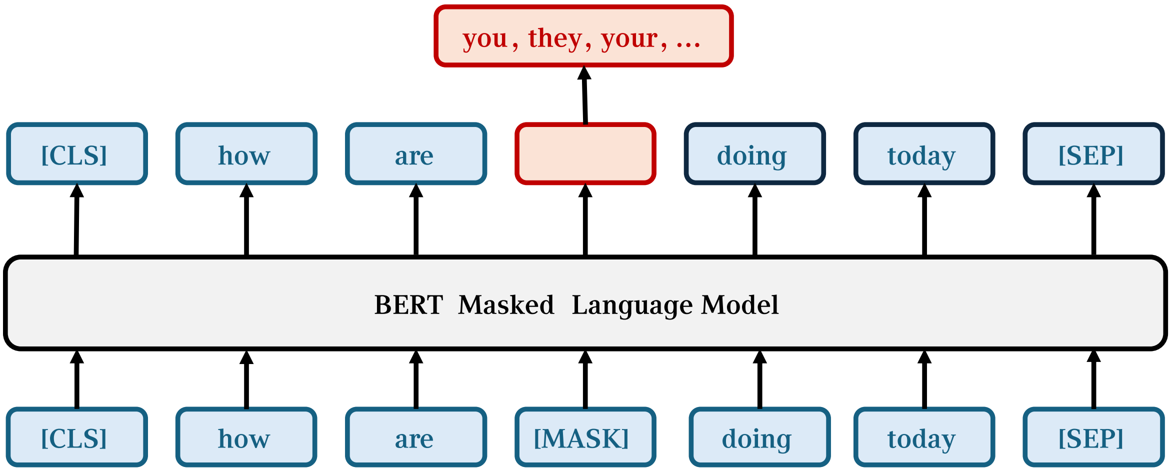


Figure 3.4: A BERT masked language model learning to predict the correct token for “you” in the sentence ”how are you doing today”.

Model	Obj.	Data	Data Size	Sizes	Chap.
BERT (Devlin et al., 2019)	MLM, NSP	Books, Wikipedia	3.3B/16GB	110M, 340M	5, 6
RoBERTa (Liu et al., 2019)	MLM	Books, Wikipedia, Additional	160GB	125M, 355M	5, 6
DeBERTa (He et al., 2021)	MLM	Books, Wikipedia, Openwebtext, Reddit, Stories	78GB	150M, 400M, 750M, 900M, 1.5B	5, 6

Table 3.1: The PLMs with MLM used in this work. Obj. stands for training objective, Chap. stands for chapter

3.2.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is a transformer encoder pre-trained with the MLM objective, with several additions. Tokens at masked positions are replaced with a [MASK] token only with an 80% chance, for the other 20%, they are either replaced with a random token, or left intact. BERT uses a next sentence prediction (NSP) objective in addition to MLM, for which the model is given the concatenation of two input sentences and has to classify if they originally appeared together or have been randomly combined. To enable to model to differentiate between the same words at different position, sinusoidal position embeddings are added to the input embeddings after the embedding layer.

BERT is trained on the BookCorpus (Zhu et al., 2015, 16GB) and an English Wikipedia dump. Two variants of BERT are released, BERT-base (110M parameters) and BERT-large (340M parameters).

3.2.2.2 RoBERTa

RoBERTa (A Robustly optimized BERT Pretraining Approach, Liu et al., 2019) uses the same architecture as BERT, but makes several improvements. The model is trained for longer and with bigger batches over more data in longer sequences, and the NSP objective is removed, as it was found to have little influence on performance. While BERT creates static masks for each sentence during the training data preparation, RoBERTa dynamically creates a new mask every time a sentence is processed. The vocabulary for RoBERTa is larger, and tokenisation is performed with byte- rather than character-level BPE (Sennrich et al., 2016). RoBERTa is trained on a 160GB mix of the BookCorpus (16GB), CC-News, the English portion of the CommonCrawl News (76GB), OpenWebText (Gokaslan and Cohen, 2019, 38GB), and Stories (Trinh and Le, 2019, 31 GB). Two model sizes are released: RoBERTA-base (125M parameters) and RoBERTa-large (355M parameters).

3.2.2.3 DeBERTa

DeBERTa (Decoding-enhanced BERT with disentangled attention, He et al., 2021) is a BERT-style model released by Microsoft that introduces two novel techniques. First, DeBERTa uses two separate vectors for the input and the contextual embedding, for which separate representations and attention matrices are computed throughout the model. Second, an enhanced mask decoder is then used to incorporate absolute positions into the decoding layer. It is trained on a similar mix of data as RoBERTa: Wikipedia[1] (12GB), BookCorpus (Zhu et al., 2015, 6GB), OpenWebText(Gokaslan and Cohen, 2019, 38GB), and Stories (Trinh and Le, 2019, 31 GB). The total data size after deduplication is 78G, much smaller than for RoBERTa. Two models are released: base (150M parameters) and large (400M parameters), xlarge (750M) (and in DeBERTa version 2, two even larger models are included: another xlarge (900M parameters) and xxlarge (1.5B parameters).

3.2.3 Large-Scale Autoregressive PLMs: Large Language Models

As the quality of encoder-only models was levelling off and more computational resources, data sources and optimisation techniques became available, the idea of decoder-only transformer models trained on autoregressive language modelling was revisited by Radford et al. (2019), with considerably greater success thanks to the larger scale of both model size and data. Models with this architecture have been demonstrated to scale effectively and are used in a new group of language models which are much larger than those based on MLM, and are therefore commonly called large language models (LLMs). The focus has therefore shifted from the paradigm of pre-training contextual representations and making them usable in downstream tasks by finetuning, to training models to generate text, and rephrasing the same downstream tasks into text generation tasks.

A compact summary of some key characteristics of the decoder-only models used in this work can be found in Table 3.2.

[1] https://dumps.wikimedia.org/enwiki/

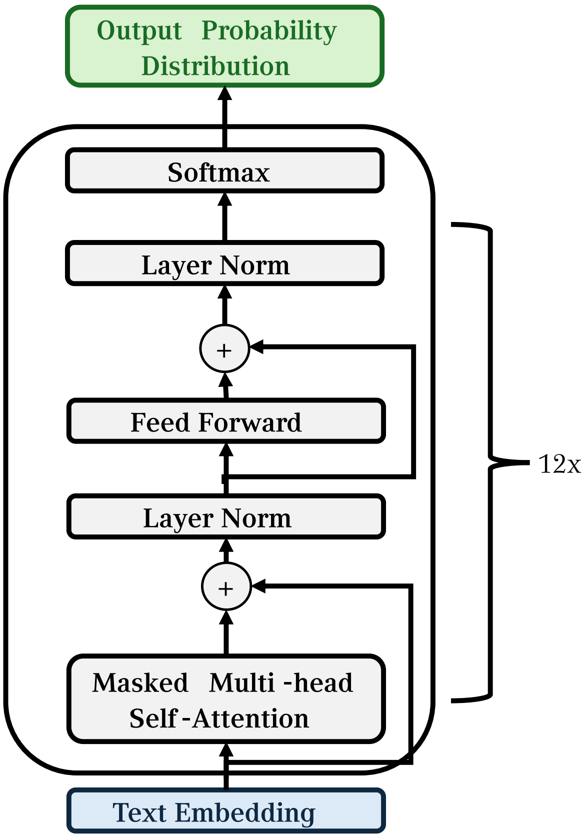

Output Probability Distribution
Softmax
Layer Norm
+
Feed Forward
Layer Norm
+
Masked Multi -head Self -Attention
Text Embedding
12x

Figure 3.5: A decoder-only transformer architecture

Model	IT	Data	Tokens	Sizes	Chap.
GPT-3.5 (Brown et al., 2020)	+	unknown	unknown	unknown	10, 9
GPT-4 (OpenAI et al., 2024)	+	unknown	unknown	unknown	11,9
GPT-J (Wang and Komatsuzaki, 2021)	−	Pile	800GB	6B	11
OPT (Zhang et al., 2022)	−	Books, Pile, Reddit	180B	125M, 350M, 1.3B, 2.7B, 6.7B, 13B, 30B, 66B, 175B	6
Llama2 (Touvron et al., 2023b)	±	unknown	2T	7B, 13B, 70B	9
Mistral (Jiang et al., 2023)	±	unknown	unknown	7B	9
Mixtral (Jiang et al., 2024)	±	unknown	unknown	8x7B	9
Gemini (Team et al., 2024)	+	unkown	unknown	1.8B, 3.25BM (Nano), unknown (Pro and Ultra)	9

Table 3.2: Large Language Models used in this work. IT stands for Instruction Tuning. ± indicates that versions with and without Instruction Tuning are available.

3.2.3.1 Instruction Tuning

Autoregressive Language Models (ALMs) are pre-trained on the task of generating a plausible continuation for a given input text, which stands in contrast to the way that users intend to interact with it. For example, given the question “Can you please give me a recipe for Carbonara?”, an ALM might simply continue with ”I had a recipe myself but I lost it”, or something similar that would have been a plausible next sentence in the training data. To counteract this issue, Instruction Tuning (IT) was introduced to specifically change the ways that users are able to interact with ALMs. IT simply means fine-tuning the model on a dataset that contains pairs of instructions and correct replies, for example “Q: Can you please give me a recipe for Carbonara? A: Sure! You will need two egg yolks ...”. This is not expected to fundamentally change the representations and model weights learned in pre-training, but rather to make the learned knowledge more accessible, in teaching the model how to interact with a user.

From the perspective of probing and evaluating LLMs, this opens up a new perspective. First, it enables us to ask the model specific questions about sentences, or the world, such as “Is something physically moving in this sentence?”, and second, it creates the exciting possibility of re-using instructions made for humans from psycholinguistic experiments and comparing the model’s behaviour to that of humans.

Some of the LLMs described in the following are only available as instruction-tuned versions (GPT-3.5, GPT-4, Gemini), while others are not instruction-tuned (GPT-J, OPT), or available in both variants, which enables us to make statements about the effect of IT (Llama2, Mistral, Mixtral).

3.2.3.2 GPT

The Generative Pretrained Transformer (GPT) family of decoder-only transforme-based models by OpenAI.

GPT (Radford et al., 2018) had 117M parameters and was trained on the BookCorpus (Zhu et al., 2015, 16GB). GPT-2 (Radford et al., 2019) was trained on OpenWebText (Gokaslan and Cohen, 2019, 38GB), which excludes all Wikipedia documents. It was available in the sizes of 124M, 335M, 774M, and 1.5B parameters. GPT-3 (Brown et al., 2020) had 175B parameters and was trained on a filtered CommonCrawl (410B tokens), WebText2 (19B tokens), Books1 and 2 (12B and 55B tokens), and Wikipedia (3B tokens).

Less information is available about the commercial models GPT-3.5 and GPT-4 used in this work. The parameters and training data of the models are unknown, and the models can only be accessed through an API or the OpenAI website.[2]

3.2.3.3 GPT-J

GPT-J (Wang and Komatsuzaki, 2021) is a 6B parameter open-source model with the GPT architecture. Crucially, it is the only autoregressive LM covered here for which the training data, the Pile (Gao et al., 2020), is fully available. This makes it uniquely suited to analyses comparing the training data statistics with the model behaviour, as we demonstrate in Chapter 11.

3.2.3.4 OPT

OPT (Open Pre-trained Transformer) (Zhang et al., 2022) is a set of LLMs in various sizes released by Meta, mirrorring the architecture of GPT. It was trained on a mix of the BookCorpus (Zhu et al., 2015, 16GB), Stories (Trinh and Le, 2019, 31 GB), and CCNews from the training corpus of RoBERTa (Liu et al., 2019), as well as a subset of the Pile (Gao et al., 2020), and the PushShift.io Reddit corpus (Baumgartner et al., 2020), with a total dataset size of 180B tokens. The available model sizes are 125M, 350M, 1.3B, 2.7B, 6.7B, 13B, 30B, 66B, and 175B.

3.2.3.5 Llama2

Llama2 (Touvron et al., 2023b) is a family of open LLMs in sizes 7B, 13B, and 70B released by Meta. Along with every model, an instruction-tuned “chat” model is also released. While the previous

[2] `openai.com`

version, Llama (Touvron et al., 2023a), provided details about the source of the training data, for Llama2 it is only known that the models were trained on 2T tokens.

For reference, Llama was trained on 1T or 1.4T tokens depending on the size of the model, with a make up of 67% English CommonCrawl, 15% C4, 4.5% each of Github, Wikipedia, and Gutenberg/Books3, 2.5% arXiv and 2% stackexchange.

3.2.3.6 Mistral and Mixtral

Mistral 7B (Jiang et al., 2023) is an open LLM released by MistralAI. It builds on the transformer architecture and adds sliding window attention, rolling buffer cache, and pre-fill and chunking. It is released alongside an instruction-tuned version, but no details about either the pretraining or the instruction-tuning data are available.

Mixtral 8x7B (Jiang et al., 2024) is a sparse mixture of experts model based on Mistral 7B. Each layer consists of 8 feedforward blocks, with a router module that chooses 2 out of the 8 blocks of parameters to process the inputs, and add their outputs together. Similarly to Mistral 7B, an instruction-tuned version is also released, trained on an instruction dataset with supervised fine-tuning and on a paired feedback dataset with Direct Preference Optimisation (Rafailov et al., 2023). No further details about the training data are provided.

3.2.3.7 Gemini

The Gemini (Team et al., 2024) family of models, released by Google, are a set of multimodal models, with the largest model, Ultra, reportedly competitive with GPT-4. The number of parameters, except for the smallest model, and the training data, are unknown. Its advantage over the GPT models, for an NLP practictioner, is that the API for Gemini Pro, the mid-size model, is free up to a certain daily quota, as of the time of writing.

3.3 Evaluation

A major theme of this thesis is that we do not create any new LLMs, but we are interested in studying how they work internally, for the (above described) purposes of learning more about language, and also trying to improve the models.

3.3.1 Metrics

In this section, we describe some of the metrics used in the following chapters to evaluate models, probes, and pipelines.

3.3.1.1 Accuracy

Take a classification task with two classes, P (the positive class) and N (the negative class). Let TP, the true positives, be the number of correctly classified positive instances and TN, the true negatives, be the number of correctly classified negative instances. The classification errors are divided into two categories: FP, the false positives, which are the number of negative instances that have been falsely classified as positive, and FN, the false negatives, which are the number of positive instances that have been falsely classified as negative.

The simplest way to evaluate the classifier is then to compute the accuracy, the ratio of correctly classified samples (TP and TN) to the entire set:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.8}$$

However, this does not give insights into the particular ways in which a classifier might be failing, and it is also misleading for imbalanced datasets: if the dataset is sufficiently imbalanced, a high accuracy could be achieved by classifying all instances as either positive or negative.

3.3.1.2 Precision, Recall, and F-measure

We therefore utilise a set of three metrics to assess classifiers: Precision, Recall, and F-measure. Precision addresses the question: out of the instances that were classified as positive, which were actually correct? Formally,

$$Precision = \frac{TP}{TP + FP} \tag{3.9}$$

Recall addresses the question: out of the actual positive instances, which were found? Formally,

$$Recall = \frac{TP}{TP + FN} \tag{3.10}$$

An overall measure for the performance of the classifier is given by the F-measure or F1-score, the harmonic mean of precision and recall. Formally,

$$F_1 = \frac{2}{Recall^{-1} + Precision^{-1}} = 2\frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{3.11}$$

3.3.1.3 Top-n-accuracy

When evaluating the prediction of a model against a ranked list of gold labels, for example, a frequency distribution of human answers, we use Top-n-accuracy, also later referred to as accuracy@k. Top-n-Accuracy is defined as the percentage of model answers that are contained in the list of the top n gold labels.

3.3.1.4 Perplexity

To evaluate how well an ALM has learned to model a sentence, we measure its probability to predict the sentence word for word, which can also be considered the probability that the model assigns to the sentence. The commonly used metric is perplexity, which is the inverse probability of the sentence. Formally, let m be a language model and $s = s_1, ..., s_n$ a sequence of words, the perplexity of m on s is defined as

$$PPL(m, s_1, ..., s_n) = 2^{-\frac{1}{n}\log_2 m(s_1,...,s_n)} \tag{3.12}$$

3.3.1.5 Pearson Correlation Coefficient

The Pearson Correlation Coefficient is used to measure the correlation between two variables. Let X and Y be the variables, the correlation is then defined as

$$\rho_{X,Y} = \frac{cov(X, Y)}{\sigma_x \sigma_Y} \tag{3.13}$$

where cov is the covariance and σ is the standard deviation of each variable. At $\rho_{X,Y} = 1$, the variables have a strong positive relationship, at $\rho_{X,Y} = 0$, they are independent, and at $\rho_{X,Y} = -1$, they have a strong negative relationship.

3.3.1.6 Entropy

The entropy of a random variable is the average level of uncertainty attached to its possible outcomes. Given a discrete random variable X, which takes values in $\mathcal{X}$ and is distributed according to $p : \mathcal{X} \to [0, 1]$, the entropy is defined as

$$H(X) := -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{3.14}$$

3.3.2 Interpretability of LLMs

Some of the work in this thesis is concerned with making statements about the internal mechanisms of LLMs. By default, they are opaque: the models are trained on raw text in an unsupervised fashion, and while we can observe their output and measure their performance on downstream tasks, we cannot without further effort determine what exactly they have learned, how they are achieving their output, and where they might have undiscovered flaws. The field of Interpretability (Belinkov et al., 2020) aims to change this with several methods.

3.3.2.1 Probing classifiers

The idea of a probing classifier is the following. First, we generate representations from our LLM for a given input sentence or set of sentences. We then train a probing classifier on the representations to predict some property, for example word class. If the classifier is able to learn the property, we can infer that the model has learned information relevant to it. The crucial point here is that the probing classifier is a small model, e.g. a perceptron (McCulloch and Pitts, 1943), and the task is a complex one, e.g. example part-of-speech tagging. We therefore assume that if the probing classifier is able to learn the task, it has succeeded not because of its own representational capacity, but because it has learned to extract the necessary information out of the LLM representations, where it was already present. This method of probing LLMs was widespread for PLMs trained with MLM. We use probing classifiers in Chapters 5, 6 and 7.

3.3.2.2 Behavioural Testing

In comparison, autoregressive models are at a disadvantage when it comes to probing classifiers. Their left-to-right processing means that contextual embeddings of any given word will only encode information learned from words to its left. In addition, the rise of commercial models, which cannot be downloaded and for which only generated text, not the hidden states, are made available through an API (Team et al., 2024; OpenAI, 2022), has further constrained the applicability of methods that rely on contextual embeddings.

An alternative to this is behavioural testing with setups taken from psycholinguistics (Ettinger, 2020). This means using tests that have been designed to assess humans' implicit knowledge of language, such as "the restaurant owner forgot which customer the waitress had __", where either the human or the language model can now complete the sentence by giving the next word. For masked language models, the blank could even be in the middle of the sentence, and would be replaced with a masked token to be predicted, e.g. "The tea [MASK] nothing". This sort of setup is difficult for an autoregressive language model, but can be circumvented with instruction-tuned models (cf. 3.2.3.1). As models can now be expected to be able to respond to questions and instructions, it becomes possible to support implicit prompting with instructions, e.g. "Fill in the blank: They wug all the time. In fact, they __ just yesterday."

While this also makes explicit evaluation possible, for example by simply asking the model "What would be the past tense of the verb 'to wug'?", it has the disadvantage that we are relying on the model's understanding of the categories of "past tense" and "verb' as defined by standard linguistic literature, which is not necessarily identical with its ability to use them correctly in generating a sentence or solve a task. It also introduces another confounding factor as, unlike humans, models are known to change their answer significantly depending on the prompt phrasing (Ishibashi et al., 2023; Lu et al., 2022; Zhou et al., 2023; Zhao et al., 2021). We use behavioural testing in Chapters 5, 6, 7, 9, 10, and 11.

3.4 Summary

We have introduced the technical background in NLP for this work. We have first introduced the Universal Dependencies treebanks. We then explained the transformer architecture, and gave details on all the specific architectures and models used in our work. Finally, we provided an overview of the evaluation metrics and paradigms that were used. We now move on to the practical work.

Part II

Construction Grammar

Chapter 4

Corresponds to the following publication:

> **Leonie Weissweiler**, Taiqi He, Naoki Otani, David R. Mortensen, Lori Levin, Hinrich Schütze (2023). Construction Grammar Provides Unique Insight into Neural Language Models. Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023), pages 85–95, Washington, D.C.. Association for Computational Linguistics.

Declaration of Co-Authorship The relationship between CxG and NLP was regularly discussed in weekly meetings with all authors. I created the initial draft summarising the literature and formulating opinions and open research questions. All authors helped review the final draft of the paper and gave advice.

Research Context Why should the established research directions of linguistic evaluation and probing be extended in the direction of Construction Grammar? What are the unique difficulties, what are potential solutions, and how would this benefit the field?

These were the questions discussed in the Construction Grammar working group during my time as a visiting PhD student in the Linguistic Lab at the Language Technologies Institute at Carnegie Mellon University. The following position paper, presented at the first workshop on Construction Grammar and NLP, summarises the (sparse) existing literature on this topic and the higher-level insights I gained. We will argue that because constructions are an essential part of language, LMs must learn to represent all of them to be considered full language models. We further categorise the main difficulties with CxG probing as opposed to established methods, and propose solutions.

In Section 5.2.3, entitled *Universal Constructicon*, this paper previews the later community efforts to annotate constructions in UD, presented in Chapter 8.

Construction Grammar Provides Unique Insight into Neural Language Models

Leonie Weissweiler[*◇], **Taiqi He**[†], **Naoki Otani**[†],
David R. Mortensen[†], **Lori Levin**[†], **Hinrich Schütze**[*◇]
[*]Center for Information and Language Processing, LMU Munich
[◇]Munich Center of Machine Learning
[†]Language Technologies Institute, Carnegie Mellon University
`weissweiler@cis.lmu.de`
`{taiqih,notani,dmortens,lsl}@cs.cmu.edu`

Abstract

Construction Grammar (CxG) has recently been used as the basis for probing studies that have investigated the performance of large pretrained language models (PLMs) with respect to the structure and meaning of constructions. In this position paper, we make suggestions for the continuation and augmentation of this line of research. We look at probing methodology that was not designed with CxG in mind, as well as probing methodology that was designed for specific constructions. We analyse selected previous work in detail, and provide our view of the most important challenges and research questions that this promising new field faces.

1 Introduction

In this paper, we will analyse existing literature investigating how well constructions and constructional information are represented in pretrained language models (PLMs). We provide context to support the argument that this is one of the most important challenges facing Language Models (LMs) today, and provide a summary of the current open research questions and how they might be tackled.

Our paper is organised as follows: In Section 2, we explain why LMs must understand constructions to be good models of language and perform effectively on downstream tasks. In Section 3, we analyse the existing literature on non-CxG-focused probing to determine its limitations in analysing constructional knowledge. In Section 4, we summarise the existing probing work that is specific to CxG and analyse its data, methodology, and findings. In Section 5, we argue that the development of an appropriate probing methodology for constructions remains an open and important research question (§5.1), and highlight the need for data collection and annotation for facilitating this area of research (§5.2). Finally, in Section 5.4, we suggest next steps that LMs might take if CxG probing reveals fundamental problems.

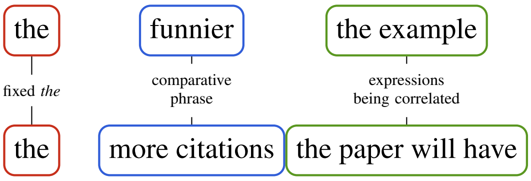


Figure 1: An example illustrating the complexity of a construction. It is an instance of the English Comparative Correlative (CC), with its syntactic features highlighted above the text and paraphrases illustrating its meaning below.

1.1 Construction Grammar

Although there are many varieties of CxG, they share the assumption that the basic building block of language structure is a pair of form and meaning. The form can be anything from a simple morpheme to the types of feature structures seen in Sign-Based Construction Grammar (SBCG) (Boas and Sag, 2012), which can be constellations of inflectional features, morphemes, categories like parts of speech, and syntactic mechanisms. Constructions with many detailed parts in SBCG include comparative constructions in sentences such as *The desk is ten inches taller than the shelf* (Hasegawa et al., 2010) and the causal excess construction as in *It was so big that it fell over* (Kay and Sag, 2012). Most importantly, the form or syntax of a sentence is not reduced to an idealized binary-branching tree or a set of hierarchically arranged pairs of head and dependants. For the purposes of this paper, we take the meaning of a construction to be a combination of Frame Semantics (Petruck and de Melo, 2014) and comparative concepts in semantics and information packaging from language typology (Croft, 2022). Because CxG does not have a clear line separating the lexicon and the grammar, the same kinds of meanings that can be associated with words can be associated with more complex structures. Table 1.1, adapted from Goldberg (2013) illustrates constructions at different

Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023), pages 85 - 95
March 9-12, 2023 ©2023 Association for Computational Linguistics

Construction Name	Construction Template	Examples
Word		Banana
Word (partially filled)	pre-N, V-ing	Pretransition, Working
Idiom (filled)		Give the devil his due
Idiom (partially filled)	Jog <someone's> memory	She jogged his memory
Idiom (minimally filled)	The X-er the Y-er	The more I think about it, the less I know
Ditransitive construction (unfilled)	Subj V Obj1 Obj2	He baked her a muffin
Passive (unfilled)	Subj aux VPpp (PP by)	The armadillo was hit by a car

Table 1: Standard examples of constructions at various levels, adapted from Goldberg (2013)

levels of complexity that contain different numbers of fixed lexemes and open slots.

In this paper, we ask whether PLMs model constructions as gestalts in both form and meaning. For example, we want to know whether a PLM represents a construction like the Comparative Correlative (*The more papers we write, the more fun we have*) as more than the sum of its individual phrases and dependencies. We also want to know whether the PLM encodes knowledge of the open slots in the construction and what can fill them. In terms of meaning, we want to find out whether the sentence's position in embedding space indicates that it has something to do with the correlation between the increase in writing more papers and having more fun. We would like to know whether PLMs represent the meaning of a correlative sentence as close to the meaning of other constructions in English and other languages that have different forms but similar meanings (e.g., *When we write more papers, we have more fun*).

1.2 Language Modelling

This paper is partially concerned with the fundamental questions of language modelling: what is its objective, and what is required of a full language model? We see the objective of language modelling very pragmatically: we aim to build a system that can predict the words in a sentence as well as possible, and therefore our aim in this paper is to point out where this requires knowledge of constructions. We do not take the objective of language modelling to mean that LMs should necessarily achieve their goal the same way that humans do. Therefore, we do not argue that language models need to "think" in terms of constructions because humans do. Rather, we consider constructions an inherent property of human language, which makes it necessary for language models to understand them.

2 Motivation

There has recently been growing interest in developing probing approaches for PLMs based on CxG. We see these approaches as coming from two different motivational standpoints, summarised below.

2.1 Constructions are Essential for Language Modelling

According to CxG, meaning is encoded in abstract constellations of linguistic units of different sizes. This means that LMs, which the field of NLP is trying to develop to achieve human language competency, must also be able to assign meaning to these units to be full LMs. Their ability to assign meaning to words, or more specifically to subword units which are sometimes closer to morphemes than to words, has been shown at length (Wiedemann et al., 2019; Reif et al., 2019; Schwartz et al., 2022). The question therefore remains: are PLMs able to retrieve and use meanings associated with patterns involving multiple tokens? We do not take this to only mean contiguous, fixed expressions, but much more importantly, non-contiguous patterns with slots that have varying constraints placed on them. To imitate and match human language behaviour, models of human language need to learn how to recognise these patterns, retrieve their meaning, apply this meaning to the context, and use them when producing language. Simply put, there is no way around learning constructions if LMs are to advance. In addition, we believe that it is an independently interesting question whether existing PLMs pick up on these abstract patterns using the current architectures and training setups, and if not, which change in architecture would be necessary to facilitate this.

2.2 Importance in Downstream Tasks

Regardless of more fundamental questions about the long-term goals of LMs, we also firmly believe that probing for CxG is relevant for analysing

Lang	Reference Translation	DeepL Translation
German	Sie nieste den Schaum von ihrem Cappuccino runter.	Sie nieste den Schaum von ihrem Cappuccino.
Italian	Lei ha starnutito via la schiuma dal suo cappuccino.	Starnutì la schiuma del suo cappuccino.
Turkish	Cappuccino'sunun köpüğünü hapşırdı.	Hapşırarak cappuccino'sunun köpüğünü uçurdu.

Table 2: Translations of 'She sneezed the foam off her cappuccino.' given by DeepL[1]. Translated back to English by humans, they all mean "She sneezed her cappuccino's foam.", which does not correctly convey the resultative meaning component, i.e., that the foam is removed from the cappuccino by the sneeze (as opposed to put there).

the challenges that face applied NLP, as evaluated on downstream tasks, at this point in time. Discussion is increasingly focusing on diagnosing the specific scenarios that are challenging for current models. Srivastava et al. (2022) propose test suites that are designed to challenge LMs, and many of them are designed by looking for 'patterns' with a non-obvious, non-literal meaning that is more than the sum of the involved words. One example of such a failure can be found in Table 2, where we provide the DeepL[1] translations for the famous instance of the caused-motion construction (Goldberg, 1995, CMC;): 'She sneezed the foam off her cappuccino', where the unusual factor is that *sneeze* does not usually take a patient argument or cause a motion. For translation, this means that it either has to use the corresponding CMC in the target language, which might be quite different in form from the English CMC, or paraphrase in a way that conveys all meaning facets. For the languages we tested, DeepL did not achieve this: the resulting sentence sounds more like the foam was sneezed onto the cappuccino, or is ambiguous between this and the correct translation. Interestingly, for Russian, the motion is conveyed in the translation, but not the fact that it is caused by a sneeze.

Targeted adversarial test suites like this translation example can be a useful resource to evaluate how well LMs perform on constructions, but more crucially, CxG theory and probing methods will inform the design of better and more systematic test suites, which in turn will be used to improve LMs (§5.4).

2.3 Diversity in Linguistics for NLP

Discussions about PLMs as models of human language processing have recently gained popularity. One forum for such discussions is the Neural Nets for Cognition Discussion Group at CogSci2022[2]. The work is still very tentative, and most people agree that LMs are not ready to be used as models of human language processing. However, the discussion about whether LMs are ready to be used as cognitive models is dominated by results of probing studies based on Generative Grammar (GG), or more specifically Transformational Grammar. This means that GG is being used as the gold standard against which the cognitive plausibility of LMs is evaluated. Studies using GG assume a direct relationship between the models' performance on probing tasks and their linguistic competency. Increased performance on GG probing tasks is seen as a sign it is becoming more reasonable to use LMs as cognitive models. Another linguistic reason for theoretical diversity is that if we could show that LMs conform better to CxG rather than GG, this might open up interesting discussions if they ever start being used as cognitive models.

3 Established Probing Methods Are Only Applicable to Some Aspects of CxG

Established probing methods have focused on different aspects of the syntactic and semantic knowledge of PLMs. In this section, we summarise the major approaches that were not designed specifically with constructions in mind. We show that although each of these methodologies deals with some aspect of CxG, and might even fully investigate some simpler constructions, none of them fully covers constructional knowledge as defined in Section 1.1.

3.1 Probing Using Contextual Embeddings

Various probing studies (Garcia et al., 2021; Chronis and Erk, 2020; Karidi et al., 2021; Yaghoobzadeh et al., 2019; *inter alia*) have focused on analysing contextual embeddings at different layers of PLMs, either of one word or multiple words, or both. The common thread in their methodology is that they compare the embeddings of the same word in different contexts, or of different words in the same context. From a constructional point of view, this requires finding two

[1] https://www.deepl.com/translator

[2] http://neural-nets-for-cognition.net

constructions with similar surface forms. By comparing the embeddings over many sentences, they are able to investigate if a certain word "knows" in which construction it is, which provides evidence for the constructional knowledge of a model.

While this is a useful starting point for probing, it is also limited. Sentences with similar constructions have to be identified, which is not always possible. More importantly, this methodology currently does not tell us anything about if the model has identified the extent of the construction correctly, or if the model has correctly learned how each slot can be filled.

3.2 Probing for Relationships Between Words

Some probing studies investigate whether a PLM recognises a word pair associated with a meaningful relationship of some kind (Rogers et al. (2020)). Most prominently, probing based on Universal Dependencies (UD; de Marneffe et al. (2021)) by Hewitt and Manning (2019) attempts to find out whether there is a high attention weight between words that are in a dependency relation where one word is the head and the other word is the dependent. They found different attention heads at different layers that seem to represent specific dependency relations such as a direct object attending to its verb, a preposition attending to its object, determiners attending to nouns, possessive pronouns attending to head nouns, and passive auxiliary verbs attending to head verbs.

The methodology as it was used by Hewitt and Manning (2019) looked at the one token that each token attended to the most. This made sense for the Hewitt and Manning (2019) study because they were probing for UD structures, which consist of binary relationships of heads and dependents in a hierarchical structure.

However, the methodology would have to be extended if we want to find out whether a whole construction with many construction elements is represented in the model in something other than a hierarchical set of binary relations. Most varieties of CxG recognise constructions with more than two daughters and constructions such as *thirty miles an hour* (Fillmore et al., 2012) in which no element is the head (headless constructions). As a research question, it is still unclear what patterns of attention we would consider as evidence that a model encodes a construction that may have headless and non-binary branches. An appropriate probing methodology has not yet been developed.

3.3 Probing with Minimal Pairs

Some works in probing based on Generative Grammar have relied on finding minimal pairs of sentences that are identical except for one specific feature that, if changed, will make the sentence ungrammatical (Wei et al., 2021). For example, in *The teacher who met the students is/*are smart*, a language model that encodes hierarchical structure would predict *is* rather than *are* after *students*, whereas a language model that was fooled by adjacency might predict *are* because it is next to *students*. The sentences can be safely compared, because only one feature, in this case, the verb being assigned the same number as the subject, is changed, and no other information can intervene or distort the probe. Other studies use a more complicated paradigm of minimal pairs involving filler-gap constructions, contrasting *I know what the lion attacked (gap) in the desert* and *I know that the lion attacked the gazelle (no gap) in the desert*.

These probing methodologies have led to productive lines of research and have been applied to complex constructions such as the Comparative Correlative Construction (Weissweiler et al., 2022). However, they depend on finding two minimally different constructions, which differ only in one way (e.g., singular/plural or gap/no gap), but close minimal pairs are simply not available for every construction.

4 CxG-specific Probing

We have argued that the most commonly used and straightforward probing methods are not sufficient for fully investigating constructional knowledge in PLMs. However, there have been several papers which have created new probing methodologies specifically for constructions. In this section, we will analyse them in terms of

- Which constructions were investigated? Does the paper investigate specific constructions or does it use a pre-compiled list of constructions or restrain itself to a subset?
- For the specific instances of their construction or constructions, what data are they using? Is it synthetic or collected from a corpus? If from a corpus, how was it collected?
- What are the key probing ideas?

Paper	Language	Source	Construction	Example
Tayyar Madabushi et al. (2020)	English	From automatically constructed list by Dunn (2017)	Personal Pronoun + didn't + V + how	We didn't know how or why.
Li et al. (2022)	English	Argument Structure Constructions according to Bencini and Goldberg (2000)	caused-motion	Bob cut the bread into the pan.
Tseng et al. (2022)	Chinese	From constructions list by (Zhan, 2017)	a + 到 + 爆, etc.	好吃到爆了！ *It's so delicious!*
Weissweiler et al. (2022)	English	McCawley (1988)	Comparative Correlative	The bigger, the better.

Table 3: Overview of constructions investigated in CxG-specific probing literature, with examples.

- Does the paper only investigate probing of (unchanged) pretrained models or is finetuning also considered?

For ease of reference, we provide an overview of the constructions investigated by each of the papers in Table 3.

4.1 CxGBERT

Tayyar Madabushi et al. (2020) investigate how well BERT (Devlin et al., 2019) can classify whether two sentences contain instances of the same construction. Their list of constructions is extracted with a modified version of Dunn (2017)'s algorithm: they induce a CxG in an unsupervised fashion over a corpus, using statistical association measures. Their list of constructions is taken directly from Dunn (2017), and they find their instances by searching for those constructions' occurrences in WikiText data. This makes the constructions possibly problematic, since they have not been verified by a linguist, which could make the conclusions drawn later from the results about BERT's handling of constructions hard to generalise from.

The key probing question of this paper is: Do two sentences contain the same construction? This does not necessarily need to be the most salient or overarching construction of the sentence, so many sentences will contain more than one instance of a construction. Crucially, the paper does not follow a direct probing approach, but rather finetunes or even trains BERT on targeted construction data, to then measure the impact on CoLA. They find that on average, models trained on sentences that were sorted into documents based on their constructions do not reliably perform better than those trained on original, unsorted data. However, they additionally test BERT Base with no additional pre-training on the task of predicting whether two sentences contain instances of the same construction, measuring accuracies of about 85% after 500 training examples for the probe. These results vary wildly depending on the frequency of the construction, which might relate back to the questionable quality of the automatically identified list of constructions.

4.2 Neural Reality of Argument Structure Constructions

Li et al. (2022) probe for LMs' handling of four argument structure constructions: ditransitive, resultative, caused-motion, and removal. Specifically, they attempt to adapt the findings of Bencini and Goldberg (2000), who used a sentence sorting task to determine whether human participants perceive the argument structure or the verb as the main factor in the overall sentence meaning. The paper aims to recreate this experiment for MiniBERTa (Warstadt et al., 2020) and RoBERTa (Liu et al., 2019), by generating sentences artificially and using agglomerative clustering on the sentence embeddings. They find that, similarly to the human data, which is sorted by the English proficiency of the participants, PLMs increasingly prefer sorting by construction as their training data size increases. Crucially, the sentences constructed for testing had no lexical overlap, such that this sorting preference must be due to an underlying recognition of a shared pattern between sentences with the same argument structure. They then conduct a second experiment, in which they insert random verbs, which are incompatible with one of the constructions, and then measure the Euclidean distance between this verb's contextual embedding and that of a verb that is prototypical for the corresponding construction. The probing idea here is that if construction information is picked up by the model, the contextual embedding of the verb should acquire some constructional meaning, which would bring it closer to the corresponding prototypical verb meaning than to the others. They indeed find that this effect is significant, for both high and low frequency verbs.

4.3 CxLM

Tseng et al. (2022) study LM predictions for the slots of various degrees of openness for a corpus of Chinese constructions. Their original data comes from a knowledge database of Mandarin Chinese constructions (Zhan, 2017), which they filter so that only constructions with a fixed repetitive element remain, which are easier to find automatically in a corpus. They filter this list down further to constructions which are rated as commonly occurring by annotators, and retrieve instances from a POS-tagged Taiwanese bulletin board corpus. They binarise the openness of a given slot in a construction and mark each word in a construction as either constant or variable. The key probing idea is then to examine the conditional probabilities that a model outputs for each type of slot, with the expectation that the prediction of variable slot words will be more difficult than that of constant ones, providing that the model has acquired some constructional knowledge. They find that this effect is significant for two different Chinese BERT-based models, as negative log-likelihoods are indeed significantly higher when predicting variable slots compared to constant ones. Interestingly, the negative log-likelihood resulting from masking the entire construction lies in the middle of the two extremes. They further evaluate a BERT-based model which is finetuned on just predicting the variable slots of the dataset they compiled and find, unsurprisingly, that this improves accuracy greatly.

4.4 Probing for the English Comparative Correlative

Weissweiler et al. (2022) investigate large PLM performance on the English Comparative Correlative (CC). There are two key probing ideas, corresponding to the investigation of the syntactic vs. the semantic component of CC. They probe for PLM understanding of CC's syntax by attempting to create minimal pairs, which consist of sentences with instances of the CC and very similar sentences which do not contain an instance of the CC. They collect minimal pairs from data by searching for sentences that fit the general pattern and manually annotate them as positive and negative instances, and additionally construct artificial minimal pairs that turn a CC sentence into a non-CC sentence by reordering words. They find that a probing classifier can distinguish between the two classes easily, using mean-pooled contextual PLM embeddings. They also probe the models' understanding of the meaning of CC, for which they choose a usage-based approach, constructing NLU-style test sentences in which an instance of the construction is given and has then to be applied in a context. They find no above-chance performance for any of the models investigated in this task.

4.5 Summary

In this section, we summarise the findings of previous work on CxG-based LM probing and analyse them in terms of the constructions that are investigated, the data that is used and the probing approaches that are applied.

4.5.1 Constructions Used

So far, Tseng et al.'s (2022) study is only the work that chose a set of constructions from a list precompiled by linguists. They constrain their selection to contain only constructions that are easy to search for in a corpus, and the resource they use only contains constructions with irregular syntax, but it is nevertheless to be considered a positive point that they are able to reach a diversity of constructions investigated. In contrast, both Li et al. (2022) and Weissweiler et al. (2022) pick one or a few constructions manually, both of which are instances of 'typical' constructions frequently discussed in the linguistic literature. This makes the work more interesting to linguists and the validity of the constructions is beyond doubt. But the downside is selection bias: the constructions that are frequently discussed are likely to have strong associated meanings and do not constitute a representative sample of constructions, from a constructions-all-the-way-down standpoint (Goldberg, 2006). Lastly, Tayyar Madabushi et al. (2020) rely on artificial data collected by Dunn (2017). We consider this method to be unreliable, but it has the resulting dataset has the advantage of variety and large scale.

4.5.2 Data Used

The two main approaches to collecting data are: (i) *patterns*: finding instances of the constructions using patterns of words / part-of-speech (POS) tags and (ii) *generation* of synthetic data. Tseng et al. (2022), Weissweiler et al. (2022) and Tayyar Madabushi et al. (2020) use patterns while Li et al. (2022) and a part of Weissweiler et al. (2022) generate data based on formal grammars. Patterns have the advantage of natural data and are less prone to accidental unwanted correlations. But there is a risk of errors in the data collection process, even after the set of constructions has to be constrained to even allow for automatic classification, and the data may have been post-corrected by manual annotation, which is time-intensive. On the other hand, generation bears challenges for making the sentences as natural as possible, which can eliminate confounding factors like lexical overlap.

4.5.3 Probing Approaches Used

Regarding the probing approaches, all previous work has had its own idea. Weissweiler et al. (2022) and Li et al. (2022) both operate on the level of sentence embeddings, classifying and clustering them respectively. Tayyar Madabushi et al. (2020) could maybe be classified with them, as it employs the Next Sentence Prediction objective (Devlin et al., 2019), which operates at the sentence level. On the other hand, another part of Weissweiler et al. (2022), as well as Tseng et al. (2022), works at the level of individual predictions for masked tokens.

The greatest difference between these works is in their concept of evidence for constructional information learned by a model, and what this information even consists of. Tayyar Madabushi et al. (2020) frame this information as ‘do these two sentences contain the same construction’, Li et al. (2022) as ‘is clustering by the construction preferred over clustering by the verb’, Weissweiler et al. (2022) as ‘can a small classifier distinguish this construction from similar-looking sentences’ and ‘can information given in form of a construction be applied in context’, and Tseng et al. (2022) as ‘are open slots more difficult to predict than closed ones’. There is little overlap to be found between these approaches, so it is difficult to draw any conclusion from more than one paper at a time.

4.5.4 Overall Findings

We nonetheless make an attempt at summarising the findings so far about large PLMs’ handling of constructional information. Regarding the structure, all findings seem to be consistent with the idea that models have picked up on the syntactic structure of constructions and recognised similarities between different instances of the same construction. This appears to hold true even when tested in different rigorous setups that exclude bias from overlapping vocabulary or accidentally similar sentence structure. This has mostly been found for English, as Tseng et al. (2022) are the only ones investigating it for a non-English language, and it remains to be seen if it holds true for lower-resources languages. Considering the acquisition of the meaning of constructions, only Weissweiler et al. (2022) have investigated this, and found no evidence that models have formed any understanding of it, but were not able to provide conclusive evidence to the contrary.

5 Research Questions

In this section, we lay out our view of the problems that are facing the emerging field of CxG-based probing and the reasons behind these challenges, and propose avenues for potential future work and improvement.

5.1 How Can We Develop Probing Methods that are a Better Fit for CxG?

Going forward, we see two directions. One is what has already been happening: keep finding new ways to get around the inherent difficulty of probing for constructions, which leads us to mostly non-conclusive and not entirely reliable evidence. The better, and more difficult way forward, is to adopt a fundamentally different methodology that would establish a standard of evidence/generalisability comparable to GG-based probing.

5.2 Data

Another reason why so little work has been done in this important field is likely the lack of data. We view the lack of data as divided into three parts: the lack of lists of constructions, the lack of meaning descriptions or even a unified meaning formalism for them, and the lack of annotated instances in corpora. We explain different opportunities for the community to obtain this data going forward below.

5.2.1 Exploiting Non-constructicon Data

Many resources are available, as already stated above, that have collected or created data with specific constructions, with the aim of making certain tasks more challenging to the models in a specific way. We can analyse those datasets and the results on them from a CxG point of view, and this can add to our pool of knowledge about what models struggle with regarding constructions. They will probably not contain any meaning descriptions, but some, like in Srivastava et al. (2022), are grouped naturally by construction, and contain instances in data, which may however be artificial.

5.2.2 Making Constructicons Available

Recently, there has been substantial work by linguists to develop constructicons for different languages (Lyngfelt et al., 2018; Ziem et al., forthcoming). Some of these constructicons are readily available online, e.g., the Brazilian Portuguese one, but many are either not available or have an interface that makes them difficult to access, e.g., because it is in the constructicon's language. Although to our knowledge, none of these constructicons contain annotated instances in text, and their meaning representations will be very difficult to unify, they are an important resource at least for lists of constructions that can be investigated by probing methods. They are especially valuable because of their linguistic diversity (English, German, Japanese, Swedish, Russian, Brazilian Portuguese), the lack of which is a major flaw in the current literature, as we stated above in §4.5.4.

5.2.3 Universal Constructicon

As a more ambitious project than simply making these constructicons available online, we firmly believe that the field would benefit greatly from an attempt to unify their representations and make them available as a shared resource. Parallels can be drawn here to UD (de Marneffe et al., 2021), a project which developed a simplified version of dependency syntax that could be universally applied and agreed upon, and then provided funding for the creation of initial resources for a range of languages, which was later greatly added to by community work in the different communities. This was a major factor in the popularisation of dependency syntax within the NLP community, to the point where it is now almost synonymous with syntax itself, due in no small part to its convenience for computational research.

As a second step after the creation of a shared online resource to access the existing constructicons, the community could consider developing a shared representation to formalise the surface form of the constructions. A dataset without meaning representation that includes multiple languages would already be a very useful resource. As a next step after that, we could think about aligning constructions across languages that encode a similar meaning. The last and most ambitious step would be unifying and linking the meaning representations, which would ideally be formalised similarly to AMR (Banarescu et al., 2013). This would enable us to develop automatic test suites that can really account for the constructions' meanings and not just their structure.

5.2.4 Annotated Instances in Text

In any stage of the development of ’construction lists’ detailed above, it would be necessary to find instances of the constructions in text. Some of the probing literature described above have generated this data artificially, which is time-consuming and also removes two important advantages of precompiled construction lists: objectivity and scale. Therefore, the ideal solution would be to find resources to have data annotated for constructions. This in itself faces many challenges from a constructions-all-the-way-down perspective: annotating even one sentence completely would be very time-consuming and require many discussions about annotation schemata in advance. A more basic way of acquiring data would be to focus on a limited set of constructions, which is selected manually, and to use pre-filtering methods similar to those employed by Tseng et al. (2022) and Weissweiler et al. (2022), to acquire simply an Inside-Outside-Beginning marking in sentences that might be instances of a construction. On the downside, this is far less linguistically rigorous and also less timeless than Universal Dependencies, which guarantees that any annotated sentence has been fully annotated and will probably not need to be revised. Nevertheless, a compromise will need to be found if annotated data is to be created at all.

5.3 CxG and Transformer Architecture

As more work is done on CxG-based probing, the field will hopefully soon be able to approach the questions that we see as crucial. Current probing techniques have not yet shown that PLMs are able to adequately handle the meaning of constructions. Assuming that more comprehensive probing techniques will show conclusively that this is not the case, is it due to a lack of data? Or is there a fundamental incompatibility of current architectures and the concept of associating a pattern with a meaning? In 5.3.1 and 5.3.2, we elaborate on why the latter might be the case.

5.3.1 Non-compositional Meaning

It is possible that constructions are intrinsically difficult for LMs because they include non-compositional meaning that is not attached to a token. It is tempting to compare them to simpler multiword expressions, which also have meaning that spans several words and that is only instantiated when they appear together. They also pose a challenge to LMs because of this, as their concept of sentence meaning is often too compositional (Liu and Neubig, 2022). The key difference is in our view, that for very complex constructions, it is not clear where in the model we can search or probe for the additional meaning.

The meaning is not attached to the words instantiating the construction, but rather to the abstract pattern itself (Croft, 2001), which we can recognise, connect mentally to previous instances and store meaning for. Once we have retrieved this meaning, it is potentially applied to the whole sentence, and can therefore have consequences for the contextual meaning of words which were never even involved in it. In a transformer-based LM, this additional meaning component cannot be stored in the static embeddings and contextualised through the attention layers, because unlike for MWEs, many constructions have very open slots, so that it is impossible to say that their meaning should somehow be stored with the meaning of the words that may instantiate them. The only place to store constructional information, therefore, remains the model weights, which are much harder to investigate or alter than the model's input, and further probing might reveal that they are unable to store it at all.

5.3.2 The Language Modelling Objective

Another possibility for fundamental difficulties arises from the nature of the training objective. PLMs are typically trained either on a masked or causal language modelling objective (Devlin et al., 2019; Radford et al., 2019). It makes sense that this incentivises them to learn word meaning in context, which they will need to predict certain words, and also relationships between words, such as simple morphological dependencies. However, information about the meaning of a construction might not often be learned in a language modelling setting, simply because it will not be needed to make the correct prediction. The meaning of a construction might not be necessary information to predict one of its component words correctly when it is masked, although its structure certainly will. In contrast, finetuning on a downstream task that requires assessment of sentence meaning, such as sentence classification, might enable us to better access the constructional meaning contained in PLMs, because the finetuning objective has required explicit use of this meaning. On the other hand, this might also be thought of as a distortion of the lens, as grammatical knowledge is not typically evaluated on finetuned models, because the findings might not generalise well.

5.4 Adapting Pretraining for CxG

If we do decide that there is a fundamental problem with the current architecture and/or training regime, the next logical step would be to think about how to alter these so that acquisition of constructional meaning becomes possible. Something similar has already been considered by Tseng et al. (2022), where models are finetuned on data that has been altered to mask entire construction instances at once, and by Tayyar Madabushi et al. (2020), which collects sentences that contain instances of the same construction into 'documents' and pretrains on them. This line of thinking, which can be summarised as data modification with constructional biases, can be further expanded, to give models some help with associating sentences with similar constructions with each other.

A far more radical idea would be to think about injecting something into the architecture that could represent this additional meaning, in the style of a position embedding, or a control token (Martin et al., 2020).

6 Conclusion

We have motivated why probing large PLMs for CxG is a very important topic both for computational linguists interested in the ideal LM and for applied NLP scientists seeking to analyse and improve the current challenges that models are facing. We then summarised and analysed the existing literature on this topic. Finally, we have given our reasons for why CxG probing remains a challenge, and detailed suggestions for further development in this field, within the realms of data, methodology, and fundamental research questions.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan

Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Giulia ML Bencini and Adele E Goldberg. 2000. The contribution of argument structure constructions to sentence meaning. *Journal of Memory and Language*, 43(4):640–651.

H. C. Boas and I. A. Sag. 2012. *Sign-Based Construction Grammar*. Center for the Study of Language and Information.

Gabriella Chronis and Katrin Erk. 2020. When is a bishop not like a rook? when it's like a rabbi! multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. Association for Computational Linguistics.

William Croft. 2001. *Radical construction grammar: Syntactic theory in typological perspective*. Oxford University Press on Demand.

William Croft. 2022. *Morphosyntax: Constructions of the World's Languages*. Cambridge University Press.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan Dunn. 2017. Computational learning of construction grammars. *Language and cognition*, 9(2):254–292.

C. J. Fillmore, R. Lee-Goldman, and R. Rhodes. 2012. The framenet constructicon. In H. C. Boas and I. A. Sag, editors, *Sign-Based Construction Grammar*. Center for the Study of Language and Information.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online. Association for Computational Linguistics.

Adele Goldberg. 2006. *Constructions at work: The nature of generalization in language*. Oxford University Press, Oxford, UK.

Adele E.. Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press.

Adele E. Goldberg. 2013. 1415 Constructionist Approaches. In *The Oxford Handbook of Construction Grammar*. Oxford University Press.

Yoko Hasegawa, Russell Lee-Goldman, Kyoko Hirose Ohara, Seiko Fujii, and Charles J Fillmore. 2010. On expressing measurement and comparison in english and japanese. *Contrastive studies in construction grammar*, 10.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Taelin Karidi, Yichu Zhou, Nathan Schneider, Omri Abend, and Vivek Srikumar. 2021. Putting words in BERT's mouth: Navigating contextualized vector spaces with pseudowords. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10300–10313, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Paul Kay and Ivan A. Sag. 2012. Cleaning up the big mess: Discontinuous dependencies and complex determiners. In H. C. Boas and I. A. Sag, editors, *Sign-Based Construction Grammar*. Center for the Study of Language and Information.

Bai Li, Zining Zhu, Guillaume Thomas, Frank Rudzicz, and Yang Xu. 2022. Neural reality of argument structure constructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7410–7423, Dublin, Ireland. Association for Computational Linguistics.

Emmy Liu and Graham Neubig. 2022. Are representations built from the ground up? an empirical examination of local composition in language models. *arXiv preprint arXiv:2210.03575*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago Timponi Torrent. 2018. *Constructicography: Constructicon development across languages*, volume 22. John Benjamins Publishing Company.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the Twelfth Language*

Resources and Evaluation Conference, pages 4689–4698, Marseille, France. European Language Resources Association.

James D McCawley. 1988. The comparative conditional construction in english, german, and chinese. In *Annual Meeting of the Berkeley Linguistics Society*, volume 14, pages 176–187.

Miriam R. L. Petruck and Gerard de Melo, editors. 2014. *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*. Association for Computational Linguistics, Baltimore, MD, USA.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Lane Schwartz, Coleman Haley, and Francis Tyers. 2022. How to encode arbitrarily complex morphology in word embeddings, no corpus needed. In *Proceedings of the first workshop on NLP applications to field linguistics*, pages 64–76, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets construction grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yu-Hsiang Tseng, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. CxLM: A construction and context-aware language model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6361–6369, Marseille, France. European Language Resources Association.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your syntax, the better your semantics? probing pretrained language models for the English comparative correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint arXiv:1909.10430*.

Yadollah Yaghoobzadeh, Katharina Kann, T. J. Hazen, Eneko Agirre, and Hinrich Schütze. 2019. Probing for semantic classes: Diagnosing the meaning content of word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5740–5753, Florence, Italy. Association for Computational Linguistics.

Weidong Zhan. 2017. On theoretical issues in building a knowledge database of chinese constructions. *Journal of Chinese Information Processing*, 31(1):230–238.

Alexander Ziem, Alexander Willich, and Sascha Michel. forthcoming. *Constructing constructicons*. John Benjamins Publishing Company.

Chapter 5

Corresponds to the following publication:

> **Leonie Weissweiler**, Valentin Hofmann, Abdullatif Köksal, Hinrich Schütze (2022). The Better Your Syntax, the Better Your Semantics? Probing Pretrained Language Models for the English Comparative Correlative. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Declaration of Co-Authorship Valentin Hofmann and I conceived the idea of testing PLMs for the Comparative Correlative. We developed the idea into an experimental setup for syntax with together with my advisor Hinrich Schütze and I implemented and conducted the experiments. Abdullatif Köksal contributed ideas for the semantics setup and advised me on the use of calibration. I implemented and conducted the semantics experiments. I wrote the initial draft of the papers. All authors helped review the final draft of the paper and gave advice.

Research Context The following paper represents our first practical work in evaluating PLMs for CxG. We chose the English Comparative Correlative as a very well-documented and multi-faceted construction. It has clearly identifiable syntactic and semantic components, as well as repetitive lexical elements that allowed us to significantly reduce the burden of annotating data by pre-filtering. This approach, implemented here with a regular expression followed immediately by manual annotation, will later be developed into a complex hybrid annotation system in Chapter 9. We also develop our first implicit test for the understanding of a construction, an approach that will be modified and used in Chapters 6, 7 and 9.

The Better Your Syntax, the Better Your Semantics? Probing Pretrained Language Models for the English Comparative Correlative

Leonie Weissweiler[*◇], Valentin Hofmann[†*], Abdullatif Köksal[*◇], Hinrich Schütze[*◇]

[*]Center for Information and Language Processing, LMU Munich
[◇]Munich Center of Machine Learning
[†]Faculty of Linguistics, University of Oxford
{weissweiler,akoksal}@cis.lmu.de
valentin.hofmann@ling-phil.ox.ac.uk

Abstract

Construction Grammar (CxG) is a paradigm from cognitive linguistics emphasising the connection between syntax and semantics. Rather than rules that operate on lexical items, it posits *constructions* as the central building blocks of language, i.e., linguistic units of different granularity that combine syntax and semantics. As a first step towards assessing the compatibility of CxG with the syntactic and semantic knowledge demonstrated by state-of-the-art pretrained language models (PLMs), we present an investigation of their capability to classify and understand one of the most commonly studied constructions, the English comparative correlative (CC). We conduct experiments examining the classification accuracy of a syntactic probe on the one hand and the models' behaviour in a semantic application task on the other, with BERT, RoBERTa, and DeBERTa as the example PLMs. Our results show that all three investigated PLMs are able to recognise the structure of the CC but fail to use its meaning. While human-like performance of PLMs on many NLP tasks has been alleged, this indicates that PLMs still suffer from substantial shortcomings in central domains of linguistic knowledge.

1 Introduction

The sentence "The better your syntax, the better your semantics." contains a construction called the English comparative correlative (CC; Fillmore, 1986). Paraphrased, it could be read as "If your syntax is better, your semantics will also be better." Humans reading this sentence are capable of doing two things: (i) *recognising* that two instances of "the" followed by an adjective/adverb in the comparative as well as a phrase of the given structure (i.e., the syntax of the CC) express a specific meaning (i.e., the semantics of the CC); (ii) *understanding* the semantic meaning conveyed by the CC, i.e., understanding that in a sentence of the given structure, the second half is somehow correlated with the first.

In this paper, we ask the following question: are pretrained language models (PLMs) able to achieve these two steps? This question is important for two reasons. Firstly, we hope that recognising the CC and understanding its meaning is challenging for PLMs, helping to set the research agenda for further improvements. Secondly, the CC is one of the most commonly studied constructions in construction grammar (CxG), a usage-based syntax paradigm from cognitive linguistics, thus providing an interesting alternative to the currently prevailing practice of analysing the syntactic capabilities of PLMs with theories from generative grammar (e.g., Marvin and Linzen, 2018).

We divide our investigation into two parts. In the first part, we examine the CC's syntactic properties and how they are represented by PLMs, with the objective to determine whether PLMs can *recognise* an instance of the CC. More specifically, we construct two syntactic probes with different properties: one is inspired by recent probing methodology (e.g., Belinkov et al., 2017; Conneau et al., 2018) and draws upon minimal pairs to quantify the amount of information contained in each PLM layer; for the other one, we write a context-free grammar (CFG) to construct approximate minimal pairs in which only the word order determines if the sentences are an instance of the CC or not. We find that starting from the third layer, all investigated PLMs are able to distinguish positive from negative instances of the CC. However, this method only covers one specific subtype of comparative sentences. To cover the full diversity of instances, we conduct an additional experiment for which we collect and manually label sentences from C4 (Raffel et al., 2020) that resemble instances of the CC, resulting in a diverse set of sentences that either are instances of the CC or resemble them closely *without* being instances of the CC. Applying the same methodology to this set of sentences, we observe that all examined PLMs are still able to separate the examples very well.

In the second part of the paper, we aim to determine if the PLMs are able to *understand* the meaning of the CC. We generate test scenarios in which a statement containing the CC is given to the PLMs, which they then have to apply in a zero-shot manner. As this way of testing PLMs is prone to a variety of biases, we introduce several mitigating methods in order to determine the full capability of the PLMs. We find that none of the PLMs we investigate perform above chance level, indicating that they are not able to understand and apply the CC in a measurable way in this context.

We make three main **contributions**:

– We present the first comprehensive study examining how well PLMs can recognise and understand a CxG construction, specifically the English comparative correlative.

– We develop a way of testing the PLMs' recognition of the CC that overcomes the challenge of probing for linguistic phenomena not lending themselves to minimal pairs.

– We adapt methods from zero-shot prompting and calibration to develop a way of testing PLMs for their understanding of the CC.[1]

2 Construction Grammar

2.1 Overview

A core assumption of generative grammar (Chomsky, 1988), which can be already found in Bloomfieldian structural linguistics (Bloomfield, 1933), is a strict separation of lexicon and grammar: grammar is conceptualized as a set of compositional and general rules that operate on a list of arbitrary and specific lexical items in generating syntactically well-formed sentences. This dichotomous view was increasingly questioned in the 1980s when several studies drew attention to the fact that linguistic units larger than lexical items (e.g., idioms) can also possess non-compositional meanings (Langacker, 1987; Lakoff, 1987; Fillmore et al., 1988; Fillmore, 1989). For instance, it is not clear how the effect of the words “let alone”(as in “she doesn't eat fish, let alone meat”) on both the syntax and the semantics of the rest of the sentence could be inferred from general syntactic rules (Fillmore et al., 1988).. This insight about the ubiquity of stored form-meaning pairings in language is adopted as the central tenet of grammatical theory by Construction Grammar (CxG; see Hoffmann and Trousdale (2013) for a comprehensive overview). Rather than a system divided into non-overlapping syntactic rules and lexical items, CxG views language as a structured system of constructions with varying granularities that encapsulate syntactic and semantic components as single linguistic signs—ranging from individual morphemes up to phrasal elements and fixed expressions (Kay and Fillmore, 1999; Goldberg, 1995). In this framework, syntactic rules can be seen as emergent abstractions over similar stored constructions (Goldberg, 2003, 2006). A different set of stored constructions can result in different abstractions and thus different syntactic rules, which allows CxG to naturally accommodate for the dynamic nature of grammar as evidenced, for instance, by inter-speaker variability and linguistic change (Hilpert, 2006).

2.2 Construction Grammar and NLP

We see three main motivations for the development of a first probing approach for CxG:

– We believe that the active discourse in (cognitive) linguistics about the best description of human language capability can be supported and enriched through a computational exploration of a wide array of phenomena and viewpoints. We think that the probing literature in NLP investigating linguistic phenomena with computational methods should be diversified to include theories and problems from all points on the broad spectrum of linguistic scholarship.

– We hope that the investigation of large PLMs' apparent capabilities to imitate human language and the mechanisms responsible for these capabilities will be enriched by introducing a usage-based approach to grammar. This is especially important as some of the discourse in recent years has focused on the question of whether PLMs are constructing syntactically acceptable sentences for the correct reasons and with the correct underlying representations (e.g. McCoy et al., 2019). We would like to suggest that considering alternative theories of grammar, specifically CxG with

[1] In order to foster research at the intersection of NLP and construction grammar, we will make our data and code available at https://github.com/LeonieWeissweiler/ComparativeCorrelative.

its incorporation of slots in constructions that may be filled by specific word types and its focus on learning without an innate, universal grammar, may be beneficial to understanding the learning process of PLMs as their capabilities advance further.

– Many constructions present an interesting challenge for PLMs. In fact, recent work in challenge datasets (Ribeiro et al., 2020) has already started using what could be considered constructions, in an attempt to identify types of sentences that models struggle with, and to point out a potential direction for improvement. One of the central tenets of CxG is the relation between the form of a construction and its meaning, or to put it in NLP terms, a model must learn to infer parts of the sentence meaning from patterns that are present in it, as opposed to words. We believe this to be an interesting challenge for future PLMs.

2.3 The English Comparative Correlative

The English comparative correlative (CC) is one of the most commonly studied constructions in linguistics, for several reasons. Firstly, it constitutes a clear example of a linguistic phenomenon that is challenging to explain in the framework of generative grammar (Culicover and Jackendoff, 1999; Abeillé and Borsley, 2008), even though there have been approaches following that school of thought (Den Dikken, 2005; Iwasaki and Radford, 2009). Secondly, it exhibits a range of interesting syntactic and semantic features, as detailed below. These reasons, we believe, also make the CC an ideal testbed for a first study attempting to extend the current trend of syntax probing for rules by developing methods for probing according to CxG.

The CC can take many different forms, some of which are exemplified here:

(1) The more, the merrier.

(2) The longer the bake, the browner the colour.

(3) The more she practiced, the better she became.

Semantically, the CC consists of two clauses, where the second clause can be seen as the dependent variable for the independent variable specified in the first one (Goldberg, 2003). It can be seen on the one hand as a statement of a general cause-and-effect relationship, as in a general conditional statement (e.g., (2) could be paraphrased as "If the bake is longer, the colour will be more brown"), and on the other as a temporal development in a comparative sentence (paraphrasing (3) as "She became better over time, and she practiced more over time"). Usage of the CC typically implies both readings at the same time. Syntactically, the CC is characterised in both clauses by an instance of "the" followed by an adverb or an adjective in the comparative, either with "-er" for some adjectives and adverbs, or with "more" for others, or special forms like "better". Special features of the comparative sentences following this are the optional omission of the future "will" and of "be", as in (1). Crucially, "the" in this construction does not function as a determiner of noun phrases (Goldberg, 2003); rather, it has a function specific to the CC and has variously been called a "degree word" (Den Dikken, 2005) or "fixed material" (Hoffmann et al., 2019).

3 Syntax

Our investigation of PLMs' knowledge of the CC is split into two parts. First, we probe for the PLMs' knowledge of the syntactic aspects of the CC, to determine if they recognise its structure. Then we devise a test of their understanding of its semantic aspects by investigating their ability to apply, in a given context, information conveyed by a CC.

3.1 Probing Methods

As the first half of our analysis of PLMs' knowledge of the CC, we investigate its syntactic aspects. Translated into probing questions, this means that we ask: can a PLM recognise an instance of the CC? Can it distinguish instances of the CC from similar-looking non-instances? Is it able to go beyond the simple recognition of its fixed parts ("The `COMP-ADJ/ADV`, the ...") and group all ways of completing the sentences that are instances of the CC separately from all those that are not? And to frame all of these questions in a syntactic probing framework: will we be able to recover, using a logistic regression as the probe, this distinguishing information from a PLM's embeddings?

The established way of testing a PLM for its syntactic knowledge has in recent years become minimal pairs (e.g., Warstadt et al., 2020, Demszky et al., 2021). This would mean pairs of sentences which are indistinguishable except for the fact that one of them is an instance of the CC and the other is not, allowing us to perfectly separate a model's knowledge of the CC from other confounding factors. While this is indeed possible for simpler syntactic phenomena such as verb-noun number agreement, there is no obvious way to construct minimal pairs for the CC. We therefore construct minimal pairs in two ways: one with artificial data based on a context-free grammar (CFG), and one with sentences extracted from C4.

3.1.1 Synthetic Data

In order to find a pair of sentences that is as close as possible to a minimal pair, we devise a way to modify the words following "The X-er" such that the sentence is no longer an instance of the construction. The pattern for a positive instance is "The ADV-er the NUM NOUN VERB", e.g., "The harder the two cats fight". To create a negative instance, we reorder the pattern to "The ADJ-er NUM VERB the NOUN", e.g., "The harder two fight the cats". The change in role of the numeral from the dependent of a head to a head itself, made possible by choosing a verb that can be either transitive or intransitive, as well as the change from an adverb to an adjective, allows us to construct a negative instance that uses the same words as the positive one, but in a different order.[2] In order to generate a large number of instances, we collect two sets each of adverbs, numerals, nouns and verbs that are mutually exclusive between training and test sets. To investigate if the model is confused by additional content in the sentences, we write an CFG to insert phrases before the start of the first half, in between the two halves, and after the second half of the CC (see Appendix, Algorithms 1 and 2 for the complete CFG).

While this setup is rigourous in the sense that positive and negative sentences are exactly matched, it comes with the drawback of only considering one type of CC. To be able to conduct a more comprehensive investigation, we adopt a complementary approach and turn to pairs extracted from C4 (see Appendix, Tables 6 and 7, for examples of training and test data). These cover a broad range of CC patterns, albeit without meeting the criterion that positive and negative samples are exactly matched.

3.1.2 Corpus-based Minimal Pairs

While accepting that positive and negative instances extracted from a corpus will automatically not be minimal and therefore contain some lexical overlap and context cues, we attempt to regularise our retrieved instances as far as possible. To form a first candidate set, we POS tag C4 using spaCy (Honnibal and Montani, 2018) and extract all sentences that follow the pattern "The" (DET) followed by either "more" and an adjective or adverb, or an adjective or adverb ending in "-er", and at any point later in the sentence again the same pattern. We discard examples with adverbs or adjectives that were falsely labelled as comparative, such as "other". We then group these sentences by their sequence of POS tags, and manually classify the sequences as either positive or negative instances. We observe that sentences sharing a POS tag pattern tend to be either all negative or all positive instances, allowing us to save annotation time by working at the POS tag pattern level instead of the sentence level. To make the final set as diverse as possible, we sort the patterns randomly and label as many as possible. In order to further reduce interfering factors in our probe, we separate the POS tag patterns between training and test sets (see Appendix, Table 8, for examples).

3.1.3 The Probe

For both datasets, we investigate the overall accuracy of our probe as well as the impact of several factors. The probe consists of training a simple logistic regression model on top of the mean-pooled sentence embeddings (Vulić et al., 2020). To quantify the impact of the length of the sentence, the start position of the construction, the position of its second half, and the distance between them, we construct four different subsets D_f^{train} and D_f^{test} from both the artificially constructed and the corpus-based dataset. For each subset, we sample sentences such that both the positive and the negative class is balanced across every value of the feature within a certain range of values. This ensures that the probes are unable to exploit correlations between a class and any of the above features. We create the dataset as follows

$$D_f = \bigcup_{v \in f_v} \bigcup_{l^* \in L} S(D, v, l^*, n^*),$$

where f is the feature, f_v is the set of values for f, $L = \{positive, negative\}$ are the labels, and S is a function that returns n^* elements from D that have value v and label l^*.

To make this task more cognitively realistic, we aim to test if a model is able to generalise from shorter sentences, which contain relat-

[2] Note that an alternative reading of this sentence exists: the numeral "two" forms the noun phrase by itself and "The harder" is still interpreted as part of the CC. The sentence is actually a positive instance on this interpretation. We regard this reading as very improbable.

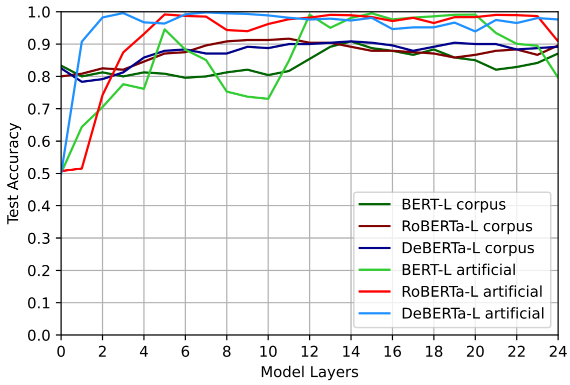


Figure 1: Overall accuracy per layer for D_{length}. All shown models are the large model variants. The models can easily distinguish between positive and negative examples in at least some of their layers.

ively little additional information besides the parts relevant to the classification task, to those with greater potential interference due to more additional content that is not useful for classification. Thus, we restrict the training set to samples from the lowest quartile of each feature so that f_v becomes $[v_f^{\min}, v_f^{\min}+\frac{1}{4}(v_f^{\max}-v_f^{\min})]$ for D_f^{train} and $[v_f^{\min}, v_f^{\max}]$ for D_f^{test}. We report the test performance for every value of a given feature separately to recognise patterns. For the artificial syntax probing, we generate 1000 data points for each value of each feature for each training and test for each subset associated with a feature. For the corpus syntax probing, we collect 9710 positive and 533 negative sentences in total, from which we choose 10 training and 5 test sentences for each value of each feature in a similar manner. To improve comparability and make the experiment computationally feasible, we test the "large" size of each of our three models, using the Huggingface Transformers library (Wolf et al., 2019). Our logistic regression probes are implemented using Scikitlearn (Pedregosa et al., 2011).

3.2 Probing Results

3.2.1 Artificial Data

As shown in Figure 1, the results of our syntactic probe indicate that all models can easily distinguish between positive and negative examples in at least some of their layers, independently of any of the sentence properties that we have investigated. We report full results in the Appendix in Figures 2, 3, and 4. We find a clear trend that DeBERTa performs better than RoBERTa, which in turn performs better than BERT across the board. As DeBERTa's performance in all layers is nearly perfect, we are unable to observe patterns related to the length of the sentence, the start position of the CC, the start position of the second half of the CC, and the distance between them. By contrast, we observe interesting patterns for BERT and RoBERTa. For D_{length}, and to a lesser degree D_{distance} (which correlates with it), we observe that at first, performance goes down with increased length as we would expect—the model struggles to generalise to longer sentences with more interference since it was only trained on short ones. However, this trend is reversed in the last few layers. We hypothesize this may be due to an increased focus on semantics in the last layers (Peters et al., 2018; Tenney et al., 2019), which could lead to interfering features particularly in shorter sentences.

3.2.2 Corpus Data

In contrast, the results of our probe on more natural data from C4 indicate two different trends: first, as the positive and negative instances are not identical on a bag-of-word level, performance is not uniformly at 50% (i.e., chance) level in the first layers, indicating that the model can exploit lexical cues to some degree. We observe a similar trend as with the artificial experiment, which showed that DeBERTa performs best and BERT worst. The corresponding graphs can be found in the Appendix in Figures 5, 6, and 7.

Generally, this additional corpus-based experiment validates our findings from the experiment with artificially generated data, as all models perform at 80% or better from the middle layers on, indicating that the models are able to classify instances of the construction even when they are very diverse and use unseen POS tag patterns.

Comparing the average accuracies on D_{length} for both data sources in Figure 1, we observe that all models perform better on artificial than on corpus data from the fifth layer on, with the notable exception of a dip in performance for BERT large around layer 10.

4 Semantics

4.1 Probing Methods

4.1.1 Usage-based Testing

For the second half of our investigation, we turn to semantics. In order to determine if a model has understood the meaning of the CC, i.e., if it has understood that in any sentence, "the `COMP` .... the

No.	Purpose	Approach	Sentence Schema
S1	Base		The ADJ1-er you are, the ADJ2-er you are. The ANT1-er you are, the ANT2-er you are. NAME1 is ADJ1-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.
S2	Bias Test	Recency	The ANT1-er you are, the ANT2-er you are. The ADJ1-er you are, the ADJ2-er you are. NAME1 is ADJ1-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.
S3		Vocabulary	The ADJ1-er you are, the ANT2-er you are. The ANT1-er you are, the ADJ2-er you are. NAME2 is ADJ1-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.
S4		Name	The ADJ1-er you are, the ADJ2-er you are. The ANT1-er you are, the ANT2-er you are. NAME2 is ADJ1-er than NAME1. Therefore, NAME2 is [MASK] than NAME1.
S5	Calibration	Short	NAME1 is ADJ1-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.
S6		Name	The ADJ1-er you are, the ADJ2-er you are. The ANT1-er you are, the ANT2-er you are. NAME1 is ADJ1-er than NAME2. Therefore, NAME3 is [MASK] than NAME4.
S7		Adjective	The ADJ1-er you are, the ADJ2-er you are. The ANT1-er you are, the ANT2-er you are. NAME1 is ADJ3-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.

Table 1: Overview of the schemata of all test scenarios used for semantic probing

COMP" implies a correlation between the two halves, we adopt a usage-based approach and ask: can the model, based on the meaning conveyed by the CC, draw a correct inference in a specific scenario? For this, we construct general test instances of the CC that consist of a desired update of the belief state of the model about the world, which we then expect it to be able to apply. More concretely, we generate sentences of the form "The ADJ1-er you are, the ADJ2-er you are.", while picking adjectives at random. To this general statement, we then add a specific scenario with two random names: "NAME1 is ADJ1-er than NAME2." and ask the model to draw an inference from it by predicting a token at the masked position in the following sentence: "Therefore, NAME1 is [MASK] than NAME2." If the model has understood the meaning conveyed by the CC and is able to use it in predicting the mask, we expect the probability of ADJ2 to be high. To provide the model with an alternative, we add a second sentence, another instance of the CC, using the antonyms of the two adjectives. This sentence is carefully chosen to have no impact on the best filler for [MASK], but also for other reasons explained in Section 4.1.2. The full test context is shown in Table 1, S1. This enables us to compare the probability of ADJ2 for the mask token directly with a plausible alternative, ANT2. One of our test sentences might be "The stronger you are, the faster you are. The weaker you are, the slower you are. Terry is stronger than John. Therefore, Terry will be [MASK] than John", where we compare the probabilities of "faster" and "slower".

Note that success in our experiment does not necessarily indicate that the model has fully understood the meaning of the CC. The experiment can only provide a lower bound for the underlying understanding of any model. However, we believe that our task is not unreasonable for a masked language model in a zero-shot setting. It is comparable in difficulty and non-reliance on world knowledge to the NLU tasks presented in LAMBADA (Paperno et al., 2016), on which GPT-2 (117M to 1.5B parameters) has achieved high zero-shot accuracy (Radford et al., Table 3). While we investigate masked language models and not GPT-2, our largest models are comparable in size to the sizes of GPT-2 that were used (340M for BERT$_{\text{L}}$, 355M for RoBERTa$_{\text{L}}$, and 1.5B parameters for DeBERTa-XXL$_{\text{L}}$), and we believe that this part of our task is achievable to some degree.

4.1.2 Biases

In this setup, we hypothesise several biases that models could exhibit and might cloud our assessment of its understanding of the CC, and devise a way to test their impact.

Firstly, we expect that models might prefer to repeat the adjective that is closest to the mask token. This has recently been documented for prompt-based experiments (Zhao et al., 2021). Here, this adjective is ANT2, the wrong answer. To test the influence this has on the prediction probabilities, we construct an alternative version of our test context in which we flip the first two sentences so that the correct answer is now more recent. The result can be found in Table 1, S2.

Secondly, we expect that models might assign higher probabilities to some adjectives, purely based on their frequency in the pretraining corpus, as for example observed by Holtzman et al. (2021). To test this, we construct a version of the test context in which ADJ2/ANT2 are swapped, which means that we can keep both the overall words the same as well as the position of the correct answer, while changing which adjective it is. The sentence is now S3 in Table 1. If there is a large difference between the prediction probabilities for the two different versions, that this means that a model's prediction is influenced by the lexical identity of the adjective in question.

Lastly, a model might have learned to associate adjectives with names in pretraining, so we construct a third version, in which we swap the names. This is S4 in Table 1. If any prior association between names and adjectives influences the prediction, we expect the scores between S4 and S1 to differ.

4.1.3 Calibration

After quantifying the biases that may prevent us from seeing a model's true capability in understanding the CC, we aim to develop methods to mitigate it. We turn to calibration, which has recently been used in probing with few-shot examples by Zhao et al. (2021). The aim of calibration is to improve the performance of a model on a classification task, by first assessing the prior probability of a label (i.e., its probability if no context is given), and then dividing the probability predicted in the task context by this prior; this gives us the conditional probability of a label given the context, representing the true knowledge of the model about this task. In adapting calibration, we want to give a model every possible opportunity to do well so that we do not underestimate its underlying comprehension.

We therefore develop three different methods of removing the important information from the context in such a way that we can use the prediction probabilities of the two adjectives in these contexts for calibration. The simplest way of doing this is to remove both instances of the CC, resulting in S5 in Table 1. If we want to keep the CC in the context, the two options to remove any information are to replace either the names or the adjectives with new names/adjectives. We therefore construct two more instances for calibration: S6 and S7 in Table 1.

For each calibration method, we collect five examples with different adjectives or names. For a given base sample S_b, we calculate P_c, the calibrated predictions, as follows:

$$P_c(a|S_b) = P(a|S_b)/[\sum_{i=1}^{i=5}(P(a|C_i)/5)]$$

where C_i is the i-th example of a given calibration technique, a is the list of adjectives tested for the masked position, and the division is applied elementwise. We collect a list of 20 adjectives and their antonyms manually from the vocabulary of the RoBERTa tokenizer and 33 common names and generate 144,800 sentences from them. We test BERT (Devlin et al., 2019) in the sizes base and large, RoBERTa (Liu et al., 2019) in the sizes base and large, and DeBERTa (He et al., 2020) in the sizes base, large, xlarge and xxlarge.

	Accuracy		Decision Flip		
	S1	S2	S2	S3	S4
BERT$_\mathrm{B}$	37.65	64.64	26.98	75.69	02.70
BERT$_\mathrm{L}$	36.85	67.21	30.44	73.31	02.32
RoBERTa$_\mathrm{B}$	61.60	52.84	09.91	76.18	02.76
RoBERTa$_\mathrm{L}$	55.71	68.00	14.33	79.47	04.33
DeBERTa$_\mathrm{B}$	49.72	49.80	00.91	99.66	01.07
DeBERTa$_\mathrm{L}$	50.88	51.40	07.04	94.83	02.23
DeBERTa$_\mathrm{XL}$	47.73	49.33	05.46	89.28	02.51
DeBERTa$_\mathrm{XXL}$	47.34	48.72	03.59	82.09	01.13

Table 2: Selected accuracies and results for the semantic probe. We report the average accuracy on the more difficult sentences in terms of recency bias (S1) and the easier ones (S2), as well as the percentage of decisions flipped by changing from the base S1 to the sentences testing for recency bias (S2), vocabulary bias (S3), and name bias (S4). RoBERTa and DeBERTa perform close to chance on S1 and S2 accuracy, indicating that they do not understand the meaning of CC. BERT's performance is strongly influenced by biases (recency, lexical identity), also indicating that it has very limited if any understanding of CC.

4.2 Results

In Table 2, we report the accuracy for all examined models. Out of the three variations to test biases, we report accuracy only for the sentence testing the recency bias as we expect this bias to occur systematically across all sentences: if it is a large effect, it will always lead to the sentence where the correct answer is the more recent one being favoured. To assess the influence of each bias beyond accuracy, we report as decision flip the percentage of sentences for which the decision (i.e., if the correct adjective had a higher probability than the incorrect one) was changed when considering the alternative sentence that was constructed to test for bias. We report full results in Appendix, Table 4.

Looking at the accuracies, we see that RoBERTa's and DeBERTa's scores are close to 50% (i.e., chance) accuracy for both S1 and S2. BERT models differ considerably as they seem to suffer from bias related to the order of the two CCs, but we can see that the average between them is also very close to chance. When we further look at the decision flips for each of the biases, we find that there is next to no bias related to the choice of names (S4). However, we can see a large bias related to both the recency of the correct answer (S2) and the choice of adjectives (S3). The recency bias is strongest in the BERT models, which also accounts for the difference in accuracies. For RoBERTa and DeBERTa models, the recency bias is small, but clearly present. In contrast, they exhibit far greater bias towards the choice of adjective, even going as far as 99.66% of decisions flipped by changing the adjective for DeBERTa base. This suggests that these models' decisions about which adjective to assign a higher probability is almost completely influenced by the choice of adjective, not the presence of the CC. Overall, we conclude that without calibration, all models seem to be highly susceptible to different combinations of bias, which completely obfuscate any underlying knowledge of the CC, leading to an accuracy at chance level across the board.

We therefore turn to our calibration methods, evaluating them first on their influence on the decision flip scores, which directly show if we were able to reduce the impact of the different types of bias. We report these only for order and vocabulary bias as we found name bias to be inconsequential. We report the complete results in Appendix, Tables 4 and 5. We see that across all models, while all three calibration methods work to reduce some bias, none does so consistently across all models or types of bias. We report the impact of all calibration methods on the final accuracies of the three largest models in Table 3. Even in cases where calibration has clearly reduced the decision flip score, we find that the final calibrated accuracy is still close to 50%. This indicates that despite the effort to retrieve any knowledge that the models have about the CC, they are unable to perform clearly above chance, and we have therefore found no evidence that the investigated models understand and can use the semantics of the CC.

Model	Test	-	S5	S6	S7
BERT$_\text{L}$	S1	36.85	31.91	47.21	44.03
	S2	67.13	73.48	54.39	64.45
	S3	36.46	43.43	47.79	44.36
RoBERTa$_\text{L}$	S1	55.72	58.37	65.08	69.53
	S2	68.01	74.53	62.73	77.76
	S3	55.36	52.02	65.28	69.23
DeBERTa$_\text{XXL}$	S1	47.35	53.56	54.92	54.12
	S2	48.73	52.85	54.03	53.81
	S3	47.57	49.36	55.25	53.59

Table 3: Effect of our three calibration methods compared to no calibration, for the three largest models. We report the accuracy scores for the base sentence (S1), recency bias (S2), and vocabulary bias (S3). The results indicate that, even if we try to address bias through calibration, the models are unable to perform clearly above chance. We have therefore found no evidence that the models understand the semantics of the CC.

4.2.1 Problem Analysis

Different conclusions might be drawn as to why none of these models have learned the semantics of the CC. They might not have seen enough examples of it in their training corpus to have formed a general understanding. Given the many examples that we were able to find in C4, and the overall positive results from the syntax section, we find this to be unlikely. Alternatively, it could be argued that models have never had a chance to learn what the CC means because they have never seen it applied, and do not have the same opportunities as humans to either interact with the speaker to clarify the meaning or to make deductions using observations in the real world. This is in line with other considerations about large PLMs acquiring advanced semantics, even though it has for many phenomena been shown that pretraining is sufficient (Radford et al., 2019). Lastly, it might be possible that the type of meaning representation required to solve this task is beyond the current transformer-style architectures. Overall, our finding that PLMs do not learn the semantics of the CC adds to the growing body of evidence that complex semantics like negation (Kassner and Schütze, 2020) is still beyond state-of-the-art PLMs.

5 Related Work

5.1 Construction Grammar in NLP

CxG has only recently and very sparsely been investigated in neural network-based NLP. Tayyar Madabushi et al. (2020) use a probe to show that while a probe on top of BERT contextual embeddings is able to mostly correctly classify if two sentences contain instances of the same construction, injecting this knowledge into the model by adding it to pretraining does not improve its performance. Our work differs from this study in that we delve deeper into what it means to understand a construction on a semantic level, and take careful precautions to isolate the recognition of the construction at the syntax level from confounding factors. Li et al. (2022) recreate the experiments of Bencini and Goldberg (2000) and Johnson and Goldberg (2013) on argument structure constructions, by creating artificial sentences with four major argument structure types and a random combination of verbs, to investigate whether PLMs prefer sorting by construction or by main verb. Tseng et al. (2022) choose items from a Chinese construction list and investigate PLM's predictions when masking the open slots, the closed slots, or the entire construction. They find that models find closed slots easier to predict than open ones. Other computational studies about CxG have either focused on automatically annotating constructions (Dunietz et al., 2017) or on the creation and evaluation of automatically built lists of constructions (Marques and Beuls, 2016; Dunn, 2019).

5.2 Probing

Our work also bears some similarity to recent work in generative grammar-based syntax probing of large PLMs in that we approximate the minimal pairs-based probing framework similar to Wei et al. (2021), Marvin and Linzen (2018) or Goldberg (2019). However, as we are concerned with different phenomena and investigating them from a different theoretical standpoint, the syntactic half of our work clearly differs.

The semantic half of our study is closest to recent work on designing challenging test cases for models such as Ribeiro et al. (2020), who design some edge cases for which most PLMs fail. Despite the different motivation, the outcome is very similar to a list of some particularly challenging constructions.

6 Conclusion

We have made a first step towards a thorough investigation of the compatibility of the paradigm of CxG and the syntactic and semantic capabilities exhibited by state-of-the-art large PLMs. For this, we chose the English comparative correlative, one of the most well-studied constructions, and investigated if large PLMs have learned it, both syntactically and semantically. We found that even though they are able to classify sentences as instances of the construction even in difficult circumstances, they do not seem to be able to extract the meaning it conveys and use it in context, indicating that while the syntactic aspect of the CC is captured in pretraining, the semantic aspect is not. We see this an indication that major future work will be needed to enable neural models to fully understand language to the same degree as humans.

Limitations

As our experimental setup requires significant customisation with regards to the properties of the specific construction we investigate, we are unable to consider other constructions or other languages in this work. We hope to be able to extend our experiments in this direction in the future. Our analysis is also limited—as all probing papers are—by the necessary indirectness of the probing tasks: we cannot directly assess the model's internal representation of the CC, but only construct tasks that might show it but are imperfect and potentially affected by external factors.

Acknowledgements

This work was funded by the European Research Council (#740516). The second author was also supported by the German Academic Scholarship Foundation. The third author was also supported by the German Federal Ministry of Education and Research (BMBF, Grant No. 01IS18036A). We thank the reviewers for their extremely helpful comments. We are also very grateful to David Mortensen and Lori Levin for helpful discussions and comments.

References

Anne Abeillé and Robert D Borsley. 2008. Comparative correlatives and parameters. *Lingua*, 118(8):1139–1157.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Giulia ML Bencini and Adele E Goldberg. 2000. The contribution of argument structure constructions to sentence meaning. *Journal of Memory and Language*, 43(4):640–651.

Leonard Bloomfield. 1933. *Language*. Holt, Rinehart & Winston, New York, NY.

Noam Chomsky. 1988. Generative grammar. *Studies in English linguistics and literature*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Peter W Culicover and Ray Jackendoff. 1999. The view from the periphery: The english comparative correlative. *Linguistic inquiry*, 30(4):543–571.

Dorottya Demszky, Devyani Sharma, Jonathan H. Clark, Vinodkumar Prabhakaran, and Jacob Eisenstein. 2021. Learning to recognize dialect features. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2021*.

Marcel Den Dikken. 2005. Comparative correlatives comparatively. *Linguistic Inquiry*, 36(4):497–532.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dunietz, Lori Levin, and Jaime Carbonell. 2017. Automatically tagging constructions of causation and their slot-fillers. *Transactions of the Association for Computational Linguistics*, 5:117–133.

Jonathan Dunn. 2019. Frequency vs. association for constraint selection in usage-based construction grammar. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 117–128, Minneapolis, Minnesota. Association for Computational Linguistics.

Charles J Fillmore. 1986. Varieties of conditional sentences. In *Eastern States Conference on Linguistics*, volume 3, pages 163–182.

Charles J. Fillmore. 1989. Grammatical construction: Theory and the familiar dichotomies. In Rainer Dietrich and Carl F. Graumann, editors, *Language processing in social context*, pages 17–38. North-Holland, Amsterdam.

Charles J. Fillmore, Paul Kay, and Mary C. O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64(3):501–538.

Adele Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, Chicago, IL.

Adele Goldberg. 2006. *Constructions at work: The nature of generalization in language*. Oxford University Press, Oxford, UK.

Adele E Goldberg. 2003. Constructions: A new theoretical approach to language. *Trends in cognitive sciences*, 7(5):219–224.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Martin Hilpert. 2006. A synchronic perspective on the grammaticalization of Swedish future constructions. *Nordic Journal of Linguistics*, 29(2):151–173.

Thomas Hoffmann, Jakob Horsch, and Thomas Brunner. 2019. The more data, the better: A usage-based account of the english comparative correlative construction. *Cognitive Linguistics*, 30(1):1–36.

Thomas Hoffmann and Graeme Trousdale. 2013. *The Oxford handbook of construction grammar*. Oxford University Press.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2018. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Eiichi Iwasaki and Andrew Radford. 2009. Comparative correlatives in english: A minimalist-cartographic analysis.

Matt A Johnson and Adele E Goldberg. 2013. Evidence for automatic accessing of constructional meaning: Jabberwocky sentences prime associated verbs. *Language and Cognitive Processes*, 28(10):1439–1452.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Paul Kay and Charles J. Fillmore. 1999. Grammatical constructions and linguistic generalizations: The *What's X doing Y?* construction. *Language*, 75(1):1–33.

George Lakoff. 1987. *Women, fire, and dangerous things: What categories reveal about the mind*. University of Chicago Press, Chicago, IL.

Ronald W. Langacker. 1987. *Foundations of cognitive grammar: Theoretical prerequisites*. Stanford University Press, Stanford, CA.

Bai Li, Zining Zhu, Guillaume Thomas, Frank Rudzicz, and Yang Xu. 2022. Neural reality of argument structure constructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7410–7423, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tânia Marques and Katrien Beuls. 2016. Evaluation strategies for computational construction grammars. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1137–1146, Osaka, Japan. The COLING 2016 Organizing Committee.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2018*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets construction grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations (ICLR) 7*.

Yu-Hsiang Tseng, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. CxLM: A construction and context-aware language model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6361–6369, Marseille, France. European Language Resources Association.

Ivan Vulić, Edoardo M. Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of* *the Association for Computational Linguistics*, 8:377–392.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Algorithm 1 Context-Free Grammar for Artificial Data Creation Training Set

```
S → SPOS | SNEG
SPOS → POS1 PUNCT POS2 ’.’ | POS1 INSERT PUNCT POS2 ’.’
SNEG → NEG1 PUNCT NEG2 ’.’ | NEG1 INSERT PUNCT NEG2 ’.’
PUNCT → ’,’ | ’;’ | ”
CORE_POS → ADV_I ’the’ NUM NOUN VERB
CORE_NEG → ADV_I NUM VERB ’the’ NOUN
POS_UPPER → ’0 The’ CORE_POS
POS_LOWER → ’0 the’ CORE_POS
NEG_UPPER → ’0 The’ CORE_NEG
NEG_LOWER → ’0 the’ CORE_NEG
POS1 → POS_UPPER | POS_UPPER ADD | START POS_LOWER | START POS_LOWER ADD
POS2 → POS_LOWER | POS_LOWER ADD
NEG1 → NEG_UPPER | NEG_UPPER ADD | START NEG_LOWER | START NEG_LOWER ADD
NEG2 → NEG_LOWER | NEG_LOWER ADD
INSERT → INSERT1 | INSERT2
INSERT2 → ADDITION BETWEEN_ADD_AND_SENT SENT
PRON → ’we’ | ’they’
ADDITION → ’, and by the way ,’ | ’, and I want to add that’ | ’, and’ PRON ’just want to say that’ | ’, and then’ PRON ’said that’ | ’, and then’ PRON ’said that’
SAY → ’say’ | ’think’ | ’mean’ | ’believe’
BETWEEN_ADD_AND_SENT → PRON SAY ’that’ | PRON SAY ’that’ | PRON SAY ’that’ | PRON SAY ’that’
LOC_SENT → PRON ’said this in’ LOC ’too’
LOC → CITY ’and’ LOC | CITY
CITY → ’Munich’ | ’Washington’ | ’Cologne’ | ’Prague’ | ’Istanbul’
SENT → ’this also holds in other cases’ | ’this is not always true’ | ’this is always true’ | ’this has only recently been the case’ | ’this has not always been the case’ | ’this has always been the case’
INSERT1 → ’without stopping’ | ’without a break’ | ’without a pause’ | ’uninterrupted’ |
START → ’Nowadays ,’ | ’Nowadays’ | ’Therefore ,’ | ’Therefore’ | ’We can’ CANWORD ’that’ | ’It is’ KNOWNWORD ’that’ | ’It follows that’ | ’Sometimes’ | ’Sometimes ,’ | ’It was recently announced that’ | ’People have told me that’ | ’I recently read in a really interesting book that’ | ’I have recently read in an established , well-known newspaper that’ | ’It was reported in a special segment on TV today that’
CANWORD → ’say’ | ’surmise’ | ’accept’ | ’state’
KNOWNWORD → ’clear’ | ’known’ | ’accepted’ | ’obvious’
ADD → TEMP | UNDER1 | TEMP UNDER1 | UNDER1 TEMP
ADV_I → ADV | ADV ’and’ ADV
TEMP → TEMP1 TEMP2
TEMP1 → ’before’ | ’after’ | ’during’
TEMP2 → ’the morning’ | ’the afternoon’ | ’the night’
UNDER1 → ’under the’ UNDER2
UNDER2 → ’bed’ | ’roof’ | ’sun’
VERB → ’push’ | ’attack’ | ’chase’ | ’beat’ | ’believe’ | ’boil’ | ’box’ | ’burn’ | ’call’ | ’date’
NOUN → ’lions’ | ’pandas’ | ’camels’ | ’pigs’ | ’horses’ | ’sheep’ | ’chickens’ | ’foxes’ | ’cows’ | ’deer’
ADV → ’worse’ | ’earlier’ | ’slower’ | ’deeper’ | ’bigger’ | ’smaller’ | ’flatter’ | ’weaker’ | ’stronger’ | ’louder’
NUM → ’twelve’ | ’thirteen’ | ’fourteen’ | ’fifteen’ | ’sixteen’ | ’seventeen’ | ’eighteen’ | ’nineteen’ | ’twenty’ | ’twenty-one’
```

Algorithm 2 Context-Free Grammar for Artificial Data Creation Test Set

```
S → SPOS | SNEG
SPOS → POS1 PUNCT POS2 ’.’ | POS1 INSERT PUNCT POS2 ’.’
SNEG → NEG1 PUNCT NEG2 ’.’ | NEG1 INSERT PUNCT NEG2 ’.’
PUNCT → ’,’ | ’;’ | ”
CORE_POS → ADV_I ’the’ NUM NOUN VERB
CORE_NEG → ADV_I NUM VERB ’the’ NOUN
POS_UPPER → ’0 The’ CORE_POS
POS_LOWER → ’0 the’ CORE_POS
NEG_UPPER → ’0 The’ CORE_NEG
NEG_LOWER → ’0 the’ CORE_NEG
POS1 → POS_UPPER | POS_UPPER ADD | START POS_LOWER | START POS_LOWER ADD
POS2 → POS_LOWER | POS_LOWER ADD
NEG1 → NEG_UPPER | NEG_UPPER ADD | START NEG_LOWER | START NEG_LOWER ADD
NEG2 → NEG_LOWER | NEG_LOWER ADD
INSERT → INSERT1 | INSERT2
INSERT2 → ADDITION BETWEEN_ADD_AND_SENT SENT
PRON → ’I’ | ’you’
ADDITION → ’, and by the way ,’ | ’, and I want to add that’ | ’, and’ PRON ’just want to say that’ | ’, and then’ PRON ’said that’ | ’, and then’ PRON ’said that’
SAY → ’say’ | ’think’ | ’mean’ | ’believe’
BETWEEN_ADD_AND_SENT → PRON SAY ’that’ | PRON SAY ’that’ | PRON SAY ’that’ | PRON SAY ’that’
LOC_SENT → PRON ’said this in’ LOC ’too’
LOC → CITY ’and’ LOC | CITY
CITY → ’London’ | ’New York’ | ’Berlin’ | ’Madrid’ | ’Paris’
SENT → ’this also holds in other cases’ | ’this is not always true’ | ’this is always true’ | ’this has only recently been the case’ | ’this has not always been the case’ | ’this has always been the case’
INSERT1 → ’without stopping’ | ’without a break’ | ’without a pause’ | ’uninterrupted’ |
START → ’Nowadays ,’ | ’Nowadays’ | ’Therefore ,’ | ’Therefore’ | ’We can’ CANWORD ’that’ | ’It is’ KNOWNWORD ’that’ | ’It follows that’ | ’Sometimes’ | ’Sometimes ,’ | ’It was recently announced that’ | ’People have told me that’ | ’I recently read in a really interesting book that’ | ’I have recently read in an established , well-known newspaper that’ | ’It was reported in a special segment on TV today that’
CANWORD → ’say’ | ’surmise’
KNOWNWORD → ’clear’ | ’known’
ADD → TEMP | UNDER1 | TEMP UNDER1 | UNDER1 TEMP
ADV_I → ADV | ADV ’and’ ADV
TEMP → TEMP1 TEMP2
TEMP1 → ’before’ | ’after’ | ’during’
TEMP2 → ’the day’ | ’the night’ | ’the evening’
UNDER1 → ’under the’ UNDER2
UNDER2 → ’bridge’ | ’stairs’ | ’tree’
VERB → ’slam’ | ’break’ | ’bleed’ | ’shake’ | ’smash’ | ’throw’ | ’strike’ | ’shoot’ | ’swallow’ | ’choke’
NOUN → ’cats’ | ’dogs’ | ’girls’ | ’boys’ | ’men’ | ’women’ | ’people’ | ’humans’ | ’mice’ | ’alligators’
ADV → ’faster’ | ’quicker’ | ’harder’ | ’higher’ | ’later’ | ’longer’ | ’shorter’ | ’lower’ | ’wider’ | ’better’
NUM → ’two’ | ’three’ | ’four’ | ’five’ | ’six’ | ’seven’ | ’eight’ | ’nine’ | ’ten’ | ’eleven’
```

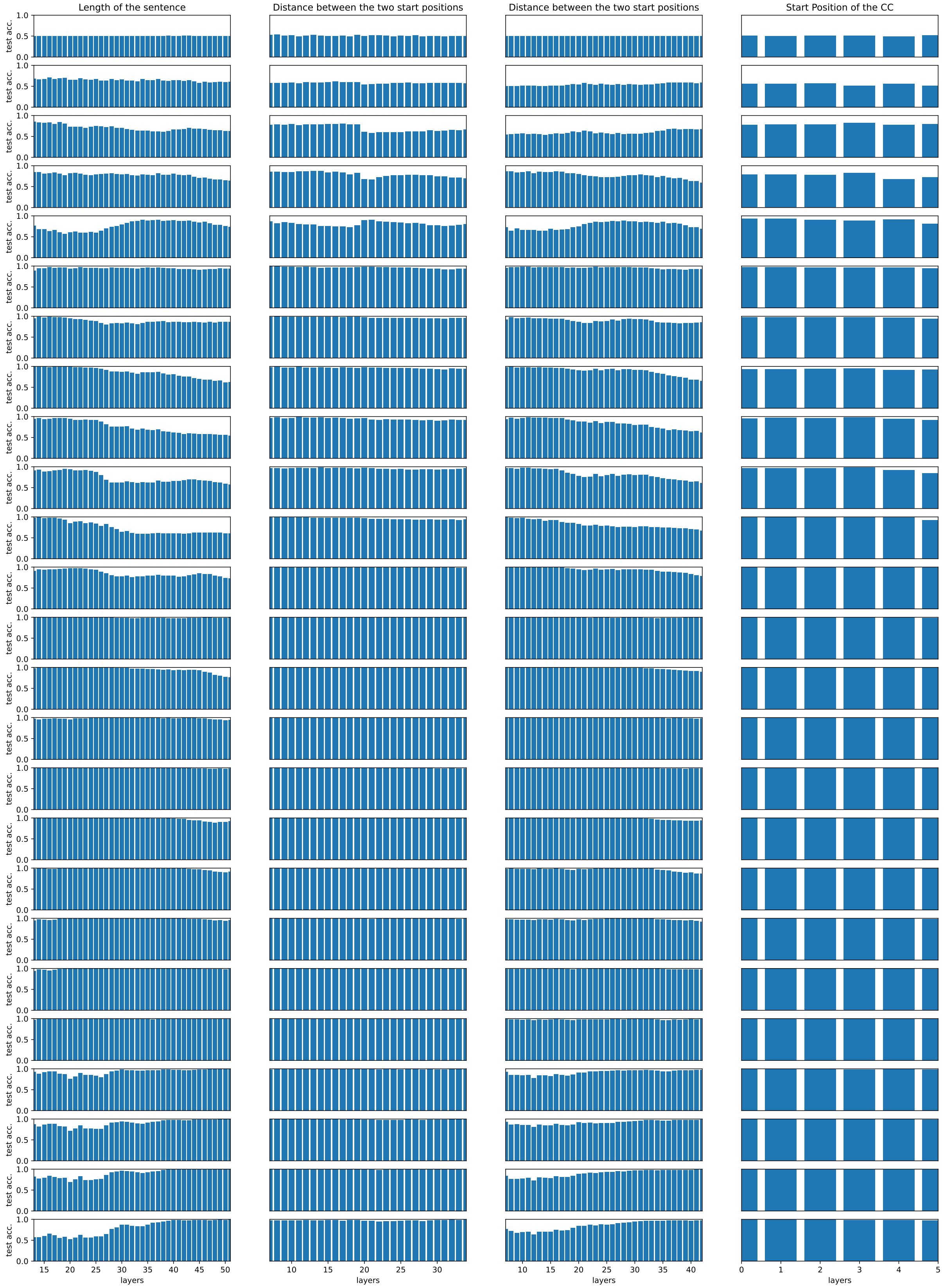


Figure 2: Full results for $\mathrm{BERT}_{\mathrm{LARGE}}$ on artificial data. Columns indicate the variable that the training and test set controls for.

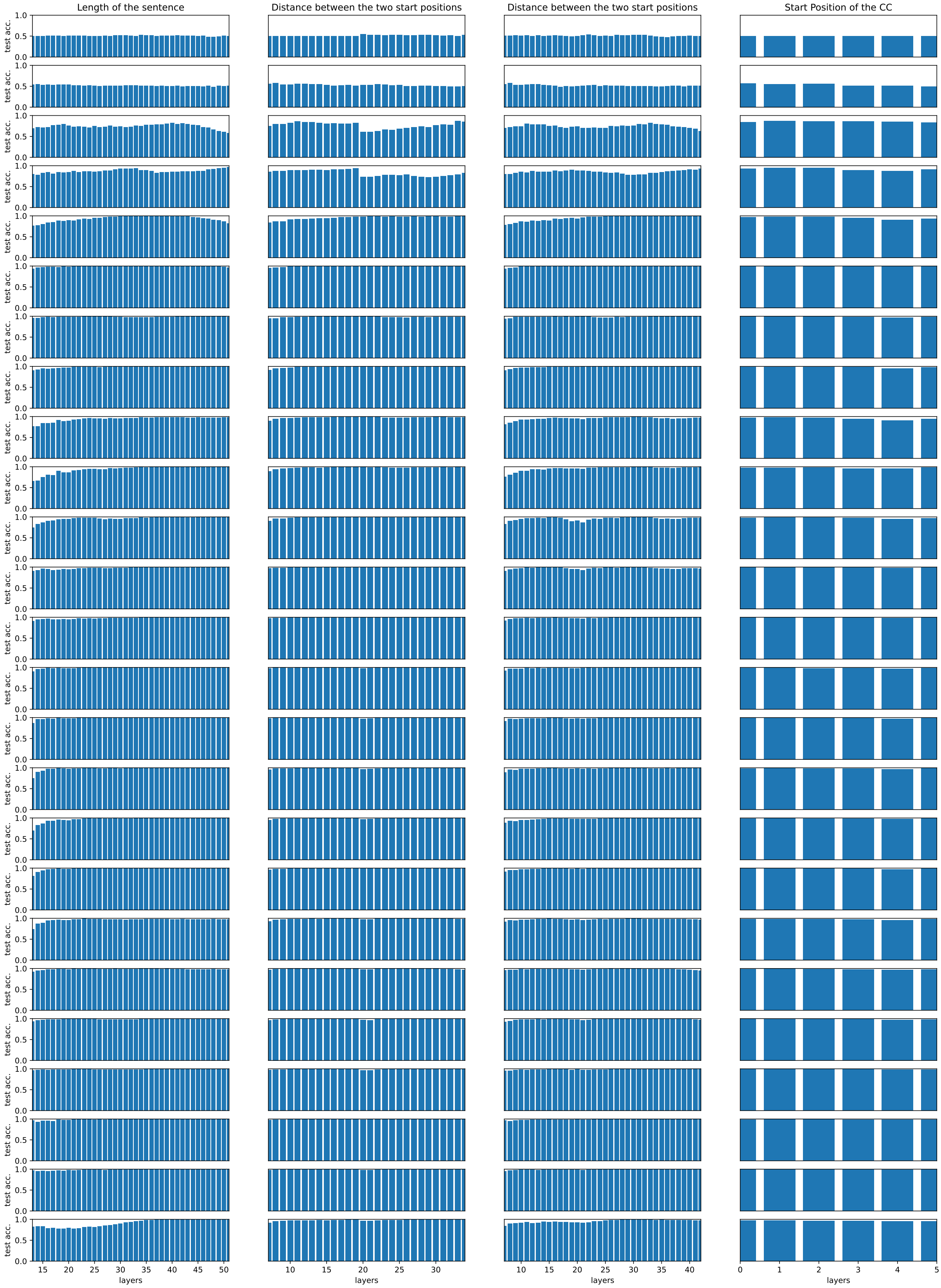


Figure 3: Full results for RoBERTa$_{\text{LARGE}}$ on artificial data. Columns indicate the variable that the training and test set controls for.

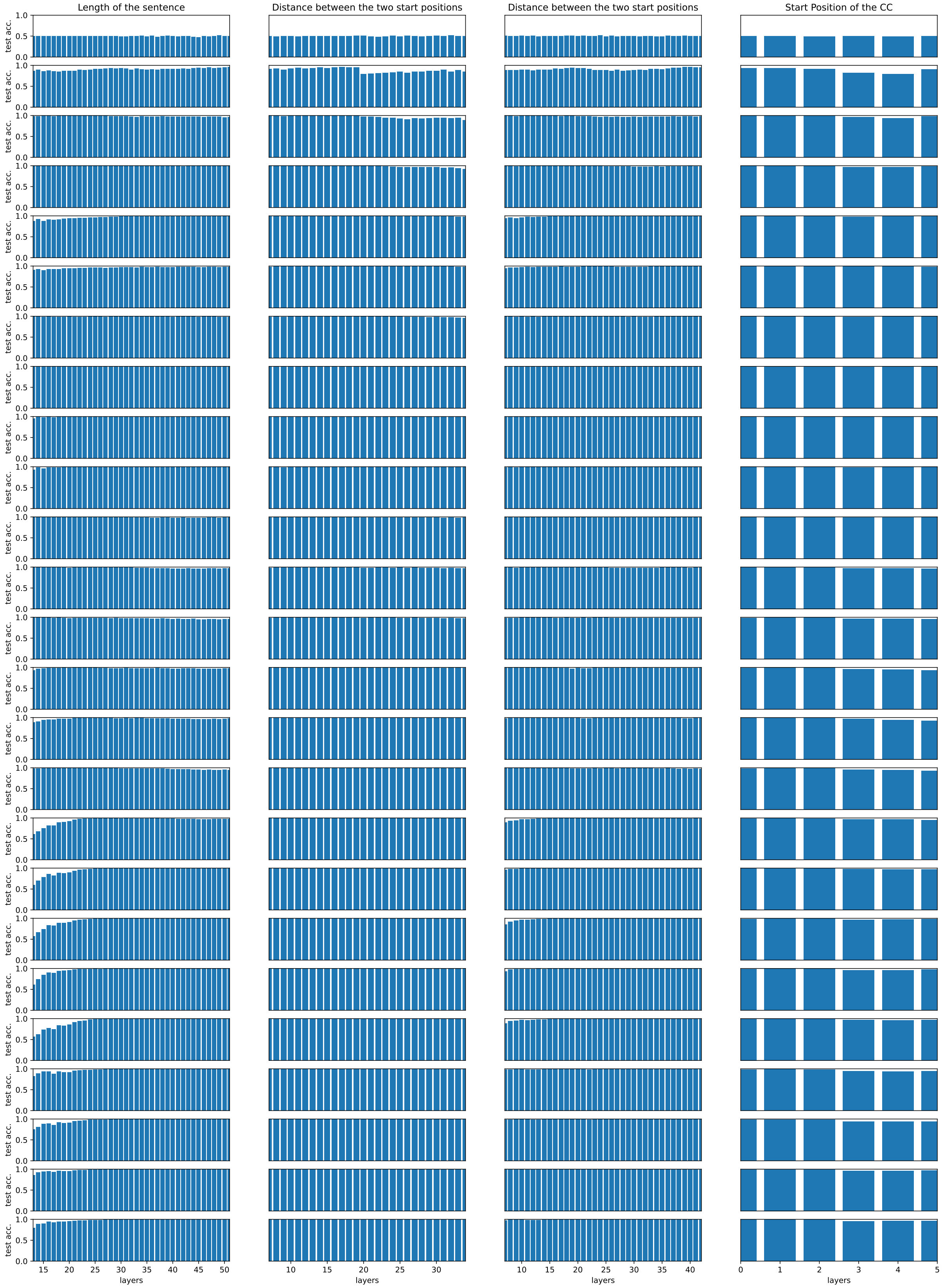


Figure 4: Full results for DeBERTa$_{\text{LARGE}}$ on artificial data. Columns indicate the variable that the training and test set controls for.

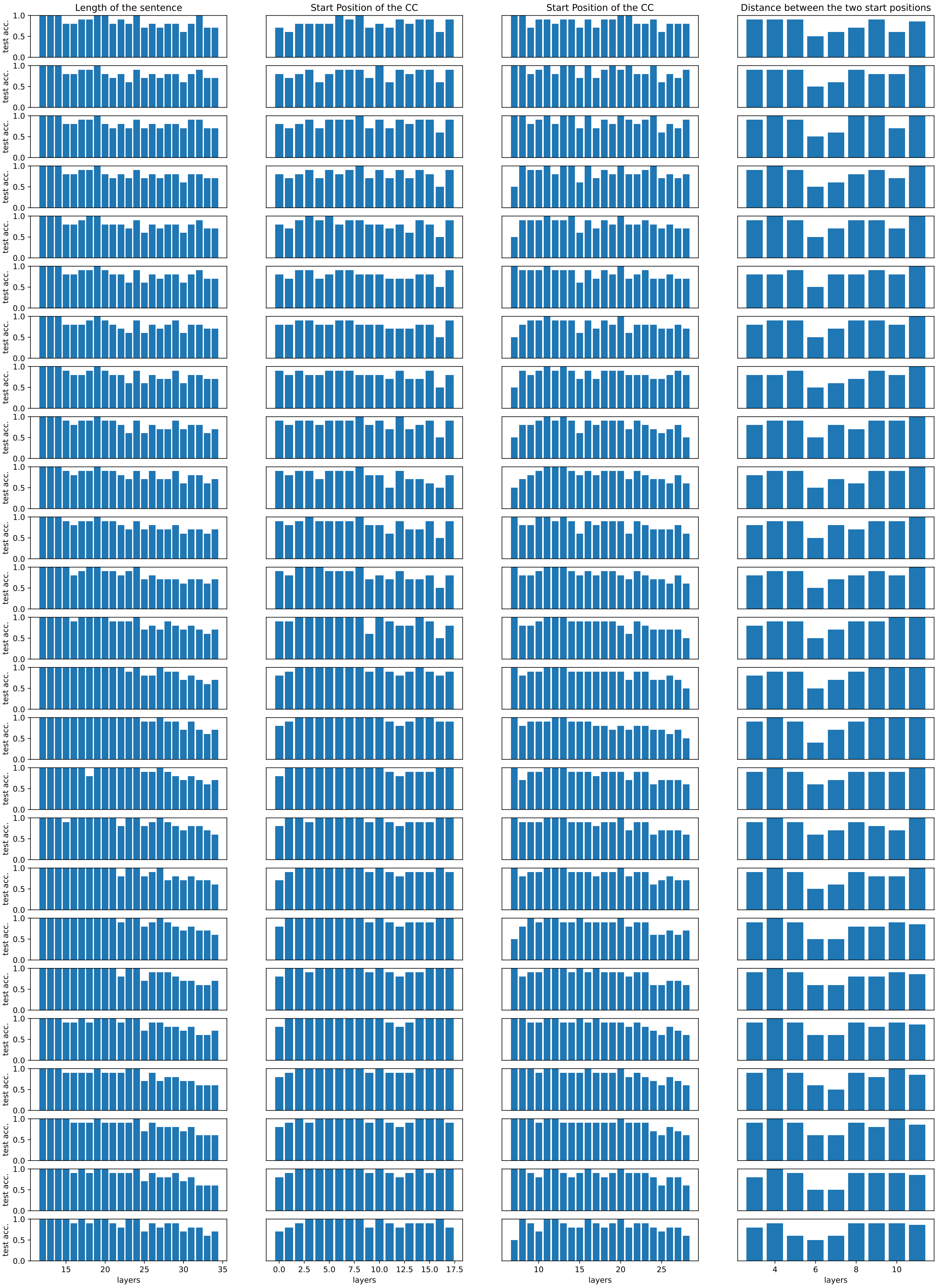


Figure 5: Full results for BERT$_{\text{LARGE}}$ on corpus data. Columns indicate the variable that the training and test set controls for.

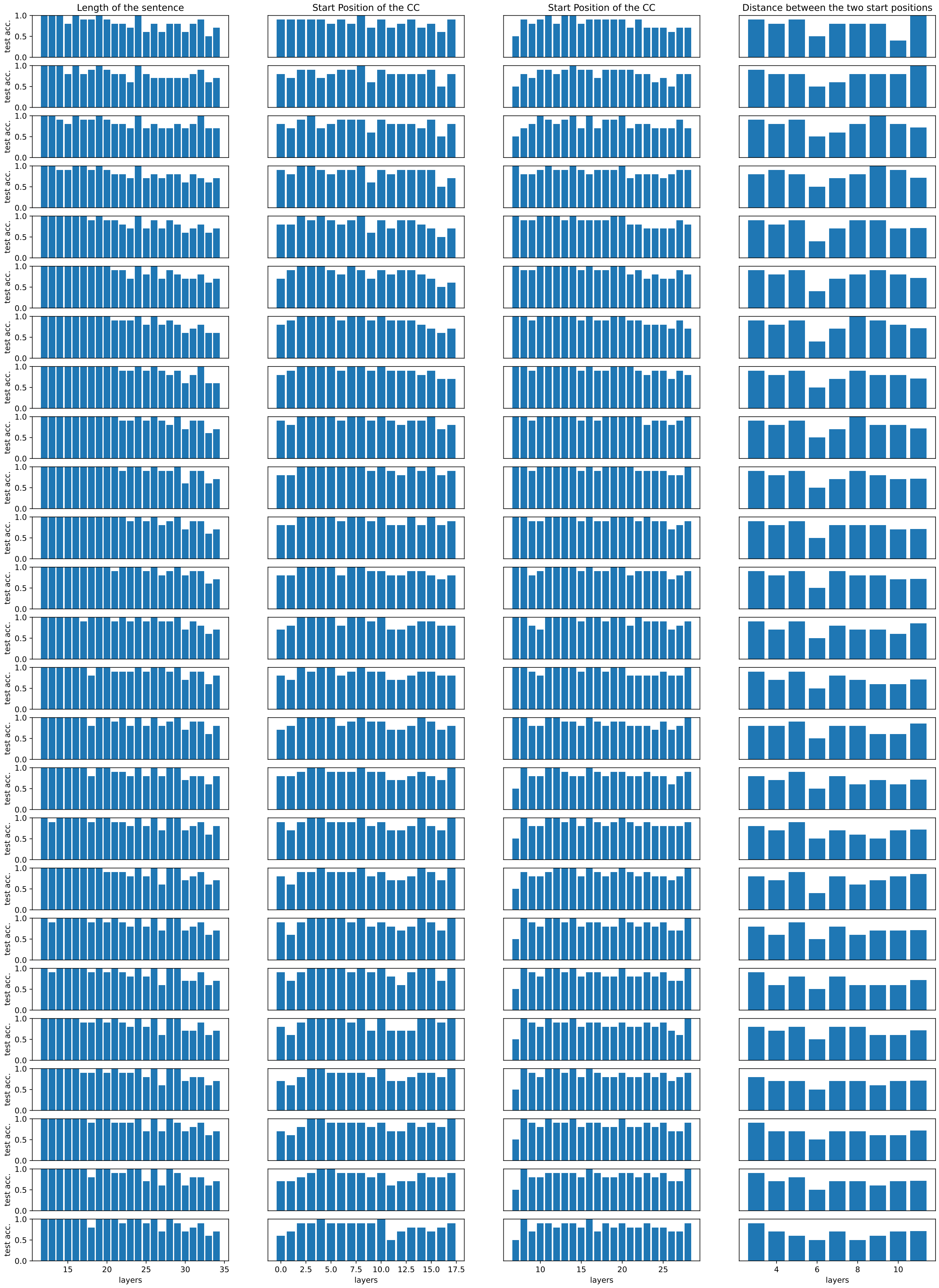


Figure 6: Full results for RoBERTa$_{\text{LARGE}}$ on corpus data. Columns indicate the variable that the training and test set controls for.

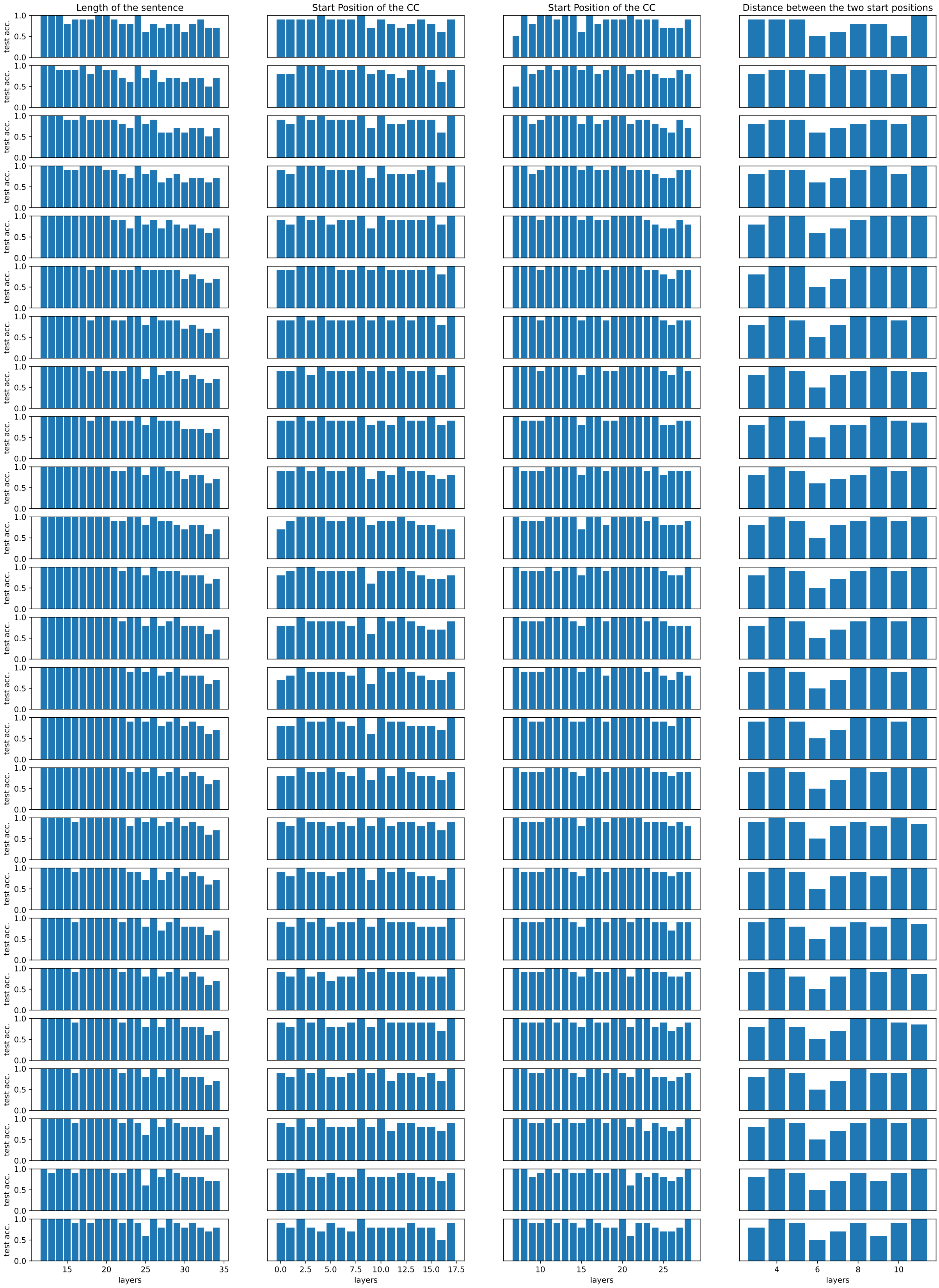


Figure 7: Full results for DeBERTa$_{\mathrm{LARGE}}$ on corpus data. Columns indicate the variable that the training and test set controls for.

Model	Test Scenario	-	S5	S6	S7
BERT_B	S1	37.65%	37.62%	44.39%	47.9%
	S2	64.64%	62.79%	56.66%	55.41%
	S3	38.04%	44.78%	44.09%	48.29%
BERT_L	S1	36.85%	31.91%	47.21%	44.03%
	S2	67.13%	73.48%	54.39%	64.45%
	S3	36.46%	43.43%	47.79%	44.36%
RoBERTa_B	S1	61.6%	58.76%	42.13%	62.32%
	S2	52.85%	51.35%	71.33%	60.25%
	S3	62.21%	55.17%	43.04%	62.76%
RoBERTa_L	S1	55.72%	58.37%	65.08%	69.53%
	S2	68.01%	74.53%	62.73%	77.76%
	S3	55.36%	52.02%	65.28%	69.23%
DeBERTa_B	S1	49.72%	49.72%	49.86%	49.2%
	S2	49.81%	48.67%	49.7%	49.06%
	S3	50.28%	50.19%	49.97%	50.0%
DeBERTa_L	S1	50.88%	49.86%	50.03%	49.39%
	S2	51.41%	48.09%	47.21%	48.04%
	S3	50.58%	49.94%	50.41%	49.42%
DeBERTa_XL	S1	47.73%	45.08%	43.31%	43.67%
	S2	49.34%	46.27%	45.58%	41.74%
	S3	47.9%	49.14%	42.68%	45.58%
$\text{DeBERTa}_\text{XXL}$	S1	47.35%	53.56%	54.92%	54.12%
	S2	48.73%	52.85%	54.03%	53.81%
	S3	47.57%	49.36%	55.25%	53.59%

Table 4: Accuracies for the semantic probe with our three calibration methods compared to no calibration. We report the average accuracy on the more difficult sentences in terms of recency bias (S1),the easier ones (S2), and vocabulary bias (S3). Our calibration tecniques are short (S5), name (S6), and adjective (S7).

Model	Test Scenario	-	S5	S6	S7
BERT$_B$	S2	26.99%	25.22%	14.75%	10.77%
	S3	75.69%	23.51%	86.33%	91.05%
	S4	2.71%	-	-	-
BERT$_L$	S2	30.44%	41.8%	13.37%	22.24%
	S3	73.31%	25.94%	88.65%	85.97%
	S4	2.32%	-	-	-
RoBERTa$_B$	S2	9.92%	8.67%	31.13%	10.86%
	S3	76.19%	22.04%	79.03%	74.75%
	S4	2.76%	-	-	-
RoBERTa$_L$	S2	14.34%	17.82%	15.94%	15.86%
	S3	79.48%	43.54%	64.78%	57.27%
	S4	4.34%	-	-	-
DeBERTa$_B$	S2	0.91%	11.77%	7.13%	10.8%
	S3	99.67%	56.44%	96.52%	94.94%
	S4	1.08%	-	-	-
DeBERTa$_L$	S2	7.04%	7.85%	14.31%	14.28%
	S3	94.83%	43.18%	85.75%	79.86%
	S4	2.24%	-	-	-
DeBERTa$_{XL}$	S2	5.47%	7.87%	13.48%	18.78%
	S3	89.28%	45.44%	68.48%	65.94%
	S4	2.51%	-	-	-
DeBERTa$_{XXL}$	S2	3.59%	3.09%	17.02%	17.21%
	S3	82.1%	79.06%	63.43%	59.81%
	S4	1.13%	-	-	-

Table 5: Decision flip scores for the semantic probe with our three calibration methods compared to no calibration. We report the percentage of decisions flipped by changing from the base S1 to the sentences testing for recency bias (S2), vocabulary bias (S3), and name bias (S4). Our calibration tecniques are short (S5), name (S6), and adjective (S7).

Sentence	Label
Nowadays , the bigger the eighteen sheep date , the louder and bigger the twelve horses beat under the sun .	Positive
The flatter the fourteen lions push , the deeper and smaller the sixteen deer burn under the roof .	Positive
The deeper the sixteen cows beat ; the flatter and earlier the twenty cows attack .	Positive
Therefore , the worse the sixteen sheep believe after the morning without a pause , the smaller the thirteen cows box after the morning under the sun .	Positive
The flatter the fourteen lions push , the deeper and smaller the sixteen deer burn under the roof .	Positive
Sometimes , the worse and earlier seventeen believe the deer , and we just want to say that they mean that this has always been the case , the flatter twenty-one attack the foxes before the afternoon under the roof .	Negative
Nowadays , the smaller sixteen box the camels , and by the way , they mean that this is always true ; the weaker thirteen date the cows .	Negative
Therefore the earlier and weaker fourteen chase the deer , the stronger and earlier thirteen boil the chickens during the night .	Negative
The weaker and worse fifteen box the lions during the morning under the sun , the worse twenty push the cows .	Negative
It follows that the worse twelve date the pigs without a break the flatter and louder nineteen call the pigs under the sun .	Negative

Table 6: Examples of artificial training data

Sentence	Label
The harder and longer the three cats throw , the harder and shorter the ten dogs shake .	Positive
I have recently read in an established , well-known newspaper that the later the ten mice strike ; the later and better the seven men smash under the tree during the night .	Positive
The shorter the ten girls break without a pause ; the later the ten boys bleed under the tree .	Positive
It was recently announced that the better and later the five women break ; the quicker the six mice smash under the tree during the evening .	Positive
The faster the seven humans choke under the stairs after the evening , and I just want to say that I think that this is not always true , the lower and higher the two boys swallow .	Positive
The higher nine strike the women without a pause the shorter ten choke the girls .	Negative
We can say that the longer and faster four strike the men under the stairs before the evening , the harder four throw the dogs after the day under the bridge .	Negative
The quicker and higher eight bleed the people , and then I said that you believe that this also holds in other cases ; the longer seven break the girls after the night .	Negative
The shorter four smash the people before the night , and by the way , you think that this is always true ; the harder three bleed the people .	Negative
The longer seven shoot the women without stopping , the faster ten strike the mice after the night under the bridge .	Negative

Table 7: Examples of artificial test data

Sentence	**Label**
" The higher up the nicer ! "	Positive
She thinks the more water she drinks the better her skin looks .	Positive
It becomes an obsession lightly because the more fish you catch the higher your adrenaline flows .	Positive
It is worth noting , however , that the more specific you are the better .	Positive
In other words , the more videos you make the greater your audience reach .	Positive
Subtract the smaller from the larger . "	Negative
The way the older guys help out the younger guys is fantastic .	Negative
In this procedure the lower lip is pulled ventrally to expose the lower incisors .	Negative
The 5th bedroom is on the lower floor with easy access to the lower bath .	Negative
Note the distinctive bend of the larger vein adjacent to the smaller vein at the top .	Negative

Table 8: Examples of corpus data

Chapter 6

Corresponds to the following publication:

> **Leonie Weissweiler**, Valentin Hofmann, Abdullatif Köksal, Hinrich Schütze (2023). Explaining Pretrained Language Models' Understanding of Linguistic Structures using Construction Grammar. Frontiers in Artificial Intelligence.

Declaration of Co-Authorship I conceived the idea of extending the experimental setup of Chapter 5 to autoregressive models together Hinrich Schütze and Valentin Hofmann. I discussed the implementation and the usage of perplexity for evaluation with Abdullatif Köksal, who also contributed code snippets. Based on these, I implemented and conducted the experiments and discussed the results with Hinrich Schütze. All authors helped review the final draft of the paper and gave advice.

Research Context This work builds on Chapter 5. Soon after the publication of the initial work on the Comparative Correlative, masked language models became less relevant and the state-of-the-art changed to autoregressive models. This raised the question: did the new models similarly struggle with representing the meaning of the Comparative Correlative? To investigate this, we adapted our methods such that similar tests could be carried out by measuring perplexity for full sentences. A desirable side-effect of the change in this setup was eliminating the need for antonym pairs, simplifying the evaluation and removing a component that could have caused the models difficulties unrelated to the construction itself.

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 12 October 2023
DOI 10.3389/frai.2023.1225791

Check for updates

OPEN ACCESS

EDITED BY
Somnath Banerjee,
University of Tartu, Estonia

REVIEWED BY
Yunfei Long,
University of Essex, United Kingdom
Anitha S. Pillai,
Hindustan Institute of Technology and Science,
India

*CORRESPONDENCE
Leonie Weissweiler
✉ weissweiler@cis.lmu.de

RECEIVED 19 May 2023
ACCEPTED 06 September 2023
PUBLISHED 12 October 2023

CITATION
Weissweiler L, Hofmann V, Köksal A and Schütze H (2023) Explaining pretrained language models' understanding of linguistic structures using construction grammar. *Front. Artif. Intell.* 6:1225791. doi: 10.3389/frai.2023.1225791

Explaining pretrained language models' understanding of linguistic structures using construction grammar

Leonie Weissweiler[1,2]*, Valentin Hofmann[1,3], Abdullatif Köksal[1,2] and Hinrich Schütze[1,2]

[1]Center for Information and Language Processing, LMU Munich, Munich, Germany, [2]Munich Center for Machine Learning, Munich, Germany, [3]Faculty of Linguistics, University of Oxford, Oxford, United Kingdom

Construction Grammar (CxG) is a paradigm from cognitive linguistics emphasizing the connection between syntax and semantics. Rather than rules that operate on lexical items, it posits *constructions* as the central building blocks of language, i.e., linguistic units of different granularity that combine syntax and semantics. As a first step toward assessing the compatibility of CxG with the syntactic and semantic knowledge demonstrated by state-of-the-art pretrained language models (PLMs), we present an investigation of their capability to classify and understand one of the most commonly studied constructions, the English comparative correlative (CC). We conduct experiments examining the classification accuracy of a syntactic probe on the one hand and the models' behavior in a semantic application task on the other, with BERT, RoBERTa, and DeBERTa as the example PLMs. Our results show that all three investigated PLMs, as well as OPT, are able to recognize the structure of the CC but fail to use its meaning. While human-like performance of PLMs on many NLP tasks has been alleged, this indicates that PLMs still suffer from substantial shortcomings in central domains of linguistic knowledge.

KEYWORDS
NLP, probing, construction grammar, computational linguistics, large language models

1. Introduction

The sentence "The better your syntax, the better your semantics." contains a construction called the English comparative correlative (CC; Fillmore, 1986). Paraphrased, it could be read as "If your syntax is better, your semantics will also be better." Humans reading this sentence are capable of doing two things: (i) *recognizing* that two instances of "the" followed by an adjective/adverb in the comparative as well as a phrase of the given structure (i.e., the syntax of the CC) express a specific meaning (i.e., the semantics of the CC); (ii) *understanding* the semantic meaning conveyed by the CC, i.e., understanding that in a sentence of the given structure, the second half is somehow correlated with the first.

In this paper, we ask the following question: are pretrained language models (PLMs) able to achieve these two steps? This question is important for two reasons. Firstly, we hope that recognizing the CC and understanding its meaning is challenging for PLMs, helping to set the research agenda for further improvements. Secondly, the CC is one of the most commonly studied constructions in construction grammar (CxG), a usage-based syntax paradigm from cognitive linguistics, thus providing an interesting alternative to the currently prevailing practice of analysing the syntactic capabilities of PLMs with theories from generative grammar (e.g., Marvin and Linzen, 2018).

We divide our investigation into two parts. In the first part, we examine the CC's syntactic properties and how they are represented by PLMs, with the objective to determine whether PLMs can *recognize* an instance of the CC. More specifically, we construct two syntactic probes with different properties: one is inspired by recent probing methodology (e.g., Belinkov et al., 2017; Conneau et al., 2018) and draws upon minimal pairs to quantify the amount of information contained in each PLM layer; for the other one, we write a context-free grammar (CFG) to construct approximate minimal pairs in which only the word order determines if the sentences are an instance of the CC or not. We find that starting from the third layer, all investigated PLMs are able to distinguish positive from negative instances of the CC. However, this method only covers one specific subtype of comparative sentences. To cover the full diversity of instances, we conduct an additional experiment for which we collect and manually label sentences from C4 (Raffel et al., 2020) that resemble instances of the CC, resulting in a diverse set of sentences that either are instances of the CC or resemble them closely *without* being instances of the CC. Applying the same methodology to this set of sentences, we observe that all examined PLMs are still able to separate the examples very well.

In the second part of the paper, we aim to determine if the PLMs are able to *understand* the meaning of the CC. We generate test scenarios in which a statement containing the CC is given to the PLMs, which they then have to apply in a zero-shot manner. As this way of testing PLMs is prone to a variety of biases, we introduce several mitigating methods in order to determine the full capability of the PLMs. We find that neither the masked language models nor the autoregressive models that we investigated performed above chance level on this task.

We make three main contributions:

- We present the first comprehensive study examining how well PLMs can recognize and understand a CxG construction, specifically the English comparative correlative.
- We develop a way of testing the PLMs' recognition of the CC that overcomes the challenge of probing for linguistic phenomena not lending themselves to minimal pairs.
- We adapt methods from zero-shot prompting and calibration to develop a way of testing PLMs for their understanding of the CC.

2. Construction grammar and natural language processing

2.1. Construction grammar

A core assumption of generative grammar (Chomsky, 1988), which can be already found in Bloomfieldian structural linguistics (Bloomfield, 1933), is a strict separation of lexicon and grammar: grammar is conceptualized as a set of compositional and general rules that operate on a list of arbitrary and specific lexical items in generating syntactically well-formed sentences. This dichotomous view was increasingly questioned in the 1980s when several studies drew attention to the fact that linguistic units larger than lexical items (e.g., idioms) can also possess non-compositional meanings (Lakoff, 1987; Langacker, 1987; Fillmore et al., 1988; Fillmore, 1989). For instance, it is not clear how the effect of the words "let alone" (as in "she doesn't eat fish, let alone meat") on both the syntax and the semantics of the rest of the sentence could be inferred from general syntactic rules (Fillmore et al., 1988). This insight about the ubiquity of stored form-meaning pairings in language is adopted as the central tenet of grammatical theory by Construction Grammar (CxG; see Hoffmann and Trousdale, 2013 for a comprehensive overview). Rather than a system divided into non-overlapping syntactic rules and lexical items, CxG views language as a structured system of constructions with varying granularities that encapsulate syntactic and semantic components as single linguistic signs—ranging from individual morphemes up to phrasal elements and fixed expressions (Goldberg A., 1995; Kay and Fillmore, 1999). In this framework, syntactic rules can be seen as emergent abstractions over similar stored constructions (Goldberg, 2003, 2006). A different set of stored constructions can result in different abstractions and thus different syntactic rules, which allows CxG to naturally accommodate for the dynamic nature of grammar as evidenced, for instance, by inter-speaker variability and linguistic change (Hilpert, 2006).

TABLE 1 **Standard examples of constructions at various levels, adapted from Goldberg (2013).**

Construction name	Construction template	Examples
Word		Banana
Word (partially filled)	pre-N, V-ing	Pretransition, Working
Idiom (filled)		Give the devil his due
Idiom (partially filled)	Jog <someone's> memory	She jogged his memory
Idiom (minimally filled)	The X-er the Y-er	The more I think about it, the less I know
Ditransitive construction (unfilled)	Subj V Obj1 Obj2	He baked her a muffin
Passive (unfilled)	Subj aux VPpp (PP by)	The armadillo was hit by a car

2.2. Why construction grammar for NLP?

There has recently been growing interest in developing probing approaches for PLMs based on CxG. We see these approaches as coming from two different motivational standpoints, summarized below.

2.2.1. Constructions are essential for language modeling

According to CxG, meaning is encoded in abstract constellations of linguistic units of different sizes. Examples of these can be found in Table 1. This means that LMs, which the field of NLP is trying to develop to achieve human language competency, must also be able to assign meaning to these units to be full LMs. Their ability to assign meaning to words, or more specifically to subword units which are sometimes closer to

TABLE 2 Translated back to English by humans, they all mean “She sneezed her cappuccino’s foam,” which does not correctly convey the resultative meaning component, i.e., that the foam is removed from the cappuccino by the sneeze (as opposed to put there).

Lang	Reference translation	DeepL translation
German	Sie nieste den Schaum von ihrem Cappuccino runter.	Sie nieste den Schaum von ihrem Cappuccino.
Italian	Lei ha starnutito via la schiuma dal suo cappuccino.	Starnutì la schiuma del suo cappuccino.
Turkish	Cappuccino’sunun köpüğünü hapşırdı.	Hapşırarak cappuccino’sunun köpüğünü uçurdu.

morphemes than to words, has been shown at length (Reif et al., 2019; Wiedemann et al., 2019; Schwartz et al., 2022). The question therefore remains: are PLMs able to retrieve and use meanings associated with patterns involving multiple tokens? We do not take this to only mean contiguous, fixed expressions, but much more importantly, non-contiguous patterns with slots that have varying constraints placed on them. To imitate and match human language behavior, models of human language need to learn how to recognize these patterns, retrieve their meaning, apply this meaning to the context, and use them when producing language. Simply put, there is no way around learning constructions if LMs are to advance. In addition, we believe that it is an independently interesting question whether existing PLMs pick up on these abstract patterns using the current architectures and training setups, and if not, which change in architecture would be necessary to facilitate this.

2.2.2. Importance in downstream tasks

Regardless of more fundamental questions about the long-term goals of LMs, we also firmly believe that probing for CxG is relevant for analysing the challenges that face applied NLP, as evaluated on downstream tasks, at this point in time. Discussion is increasingly focusing on diagnosing the specific scenarios that are challenging for current models. Srivastava et al. (2023) propose test suites that are designed to challenge LMs, and many of them are designed by looking for “patterns” with a non-obvious, non-literal meaning that is more than the sum of the involved words. One example of such a failure can be found in Table 2, where we provide the DeepL[1] translations for the famous instance of the caused-motion construction (Goldberg A. E., 1995, CMC): “She sneezed the foam off her cappuccino,” where the unusual factor is that *sneeze* does not usually take a patient argument or cause a motion. For translation, this means that it either has to use the corresponding CMC in the target language, which might be quite different in form from the English CMC, or paraphrase in a way that conveys all meaning facets. For the languages we tested, DeepL did not achieve this: the resulting sentence sounds more like the foam was sneezed onto the cappuccino, or is ambiguous between this and the correct translation. Interestingly, for Russian, the motion is conveyed in the translation, but not the fact that it is caused by a sneeze.

Targeted adversarial test suites like this translation example can be a useful resource to evaluate how well LMs perform on constructions, but more crucially, CxG theory and probing methods will inform the design of better and more systematic test suites, which in turn will be used to improve LMs.

2.2.3. Diversity in linguistics for NLP

Discussions about PLMs as models of human language processing have recently gained popularity. One forum for such discussions is the Neural Nets for Cognition Discussion Group at CogSci2022[2]. The work is still very tentative, and most people agree that LMs are not ready to be used as models of human language processing. However, the discussion about whether LMs are ready to be used as cognitive models is dominated by results of probing studies based on Generative Grammar (GG), or more specifically Transformational Grammar. This means that GG is being used as the gold standard against which the cognitive plausibility of LMs is evaluated. Studies using GG assume a direct relationship between the models’ performance on probing tasks and their linguistic competency. Increased performance on GG probing tasks is seen as a sign it is becoming more reasonable to use LMs as cognitive models. Another linguistic reason for theoretical diversity is that if we could show that LMs conform better to CxG rather than GG, this might open up interesting discussions if they ever start being used as cognitive models.

3. The English comparative correlative

The English comparative correlative (CC) is one of the most commonly studied constructions in linguistics, for several reasons. Firstly, it constitutes a clear example of a linguistic phenomenon that is challenging to explain in the framework of generative grammar (Culicover and Jackendoff, 1999; Abeillé and Borsley, 2008), even though there have been approaches following that school of thought (Den Dikken, 2005; Iwasaki and Radford, 2009). Secondly, it exhibits a range of interesting syntactic and semantic features, as detailed below. These reasons, we believe, also make the CC an ideal testbed for a first study attempting to extend the current trend of syntax probing for rules by developing methods for probing according to CxG.

The CC can take many different forms, some of which are exemplified here:

(1) The more, the merrier.
(2) The longer the bake, the browner the color.
(3) The more she practiced, the better she became.

Semantically, the CC consists of two clauses, where the second clause can be seen as the dependent variable for the independent variable specified in the first one (Goldberg, 2003). It can be seen on the one hand as a statement of a general cause-and-effect relationship, as in a general conditional statement [e.g., (2) could be paraphrased as “If the bake is longer, the color will be more brown”], and on the other as a temporal development in a comparative sentence [paraphrasing (3) as “She became better over time, and she practiced more over time”]. Usage of the CC typically implies both readings at the same time. Syntactically, the CC is characterized in both clauses by an instance of “the” followed by an adverb or an adjective in the comparative, either with “-er” for

[1] https://www.deepl.com/translator

[2] http://neural-nets-for-cognition.net

some adjectives and adverbs, or with "more" for others, or special forms like "better." Special features of the comparative sentences following this are the optional omission of the future "will" and of "be," as in (1). Crucially, "the" in this construction does not function as a determiner of noun phrases (Goldberg, 2003); rather, it has a function specific to the CC and has variously been called a "degree word" (Den Dikken, 2005) or "fixed material" (Hoffmann et al., 2019).

4. Related work

4.1. Construction grammar probing

4.1.1. CxGBERT

Tayyar Madabushi et al. (2020) investigate how well BERT (Devlin et al., 2019) can classify whether two sentences contain instances of the same construction. Their list of constructions is extracted with a modified version of Dunn (2017)'s algorithm: they induce a CxG in an unsupervised fashion over a corpus, using statistical association measures. Their list of constructions is taken directly from Dunn (2017), and they find their instances by searching for those constructions' occurrences in WikiText data. This makes the constructions possibly problematic, since they have not been verified by a linguist, which could make the conclusions drawn later from the results about BERT's handling of constructions hard to generalize from.

The key probing question of this paper is: Do two sentences contain the same construction? This does not necessarily need to be the most salient or overarching construction of the sentence, so many sentences will contain more than one instance of a construction. Crucially, the paper does not follow a direct probing approach, but rather finetunes or even trains BERT on targeted construction data, to then measure the impact on CoLA. They find that on average, models trained on sentences that were sorted into documents based on their constructions do not reliably perform better than those trained on original, unsorted data. However, they additionally test BERT Base with no additional pre-training on the task of predicting whether two sentences contain instances of the same construction, measuring accuracies of about 85% after 500 training examples for the probe. These results vary wildly depending on the frequency of the construction, which might relate back to the questionable quality of the automatically identified list of constructions.

4.1.2. Neural reality of argument structure constructions

Li et al. (2022) probe for LMs' handling of four argument structure constructions: ditransitive, resultative, caused-motion, and removal. Specifically, they attempt to adapt the findings of Bencini and Goldberg (2000), who used a sentence sorting task to determine whether human participants perceive the argument structure or the verb as the main factor in the overall sentence meaning. The paper aims to recreate this experiment for MiniBERTa (Warstadt et al., 2020b) and RoBERTa (Liu et al., 2019), by generating sentences artificially and using agglomerative clustering on the sentence embeddings. They find that, similarly to the human data, which is sorted by the English proficiency of the participants, PLMs increasingly prefer sorting by construction as their training data size increases. Crucially, the sentences constructed for testing had no lexical overlap, such that this sorting preference must be due to an underlying recognition of a shared pattern between sentences with the same argument structure. They then conduct a second experiment, in which they insert random verbs, which are incompatible with one of the constructions, and then measure the Euclidean distance between this verb's contextual embedding and that of a verb that is prototypical for the corresponding construction. The probing idea here is that if construction information is picked up by the model, the contextual embedding of the verb should acquire some constructional meaning, which would bring it closer to the corresponding prototypical verb meaning than to the others. They indeed find that this effect is significant, for both high and low frequency verbs.

4.1.3. CxLM

Tseng et al. (2022) study LM predictions for the slots of various degrees of openness for a corpus of Chinese constructions. Their original data comes from a knowledge database of Mandarin Chinese constructions (Zhan, 2017), which they filter so that only constructions with a fixed repetitive element remain, which are easier to find automatically in a corpus. They filter this list down further to constructions which are rated as commonly occurring by annotators, and retrieve instances from a POS-tagged Taiwanese bulletin board corpus. They binarize the openness of a given slot in a construction and mark each word in a construction as either constant or variable. The key probing idea is then to examine the conditional probabilities that a model outputs for each type of slot, with the expectation that the prediction of variable slot words will be more difficult than that of constant ones, providing that the model has acquired some constructional knowledge. They find that this effect is significant for two different Chinese BERT-based models, as negative log-likelihoods are indeed significantly higher when predicting variable slots compared to constant ones. Interestingly, the negative log-likelihood resulting from masking the entire construction lies in the middle of the two extremes. They further evaluate a BERT-based model which is finetuned on just predicting the variable slots of the dataset they compiled and find, unsurprisingly, that this improves accuracy greatly.

4.1.4. A discerning several thousand judgments

Mahowald (2023) focuses on the English Article + Adjective + Numeral + Noun (AANN) construction, e.g. "The president has had a terrible 5 weeks" and GPT-3's recognition of its particular semantic and syntactic constraints. He designs a few-shot prompt for grammatical acceptability using the CoLA corpus of linguistic acceptability (Warstadt et al., 2019). As probing data, he artificially constructs several variants of the AANN construction to test for GPT-3's understanding of its properties. Its output on the linguistic acceptability task is also contrasted with human ratings sourced from Mechanical Turk. The probing concept exploits that the AANN construction has several properties that seem to violate a number of rules: "a" is not marking a singular here, as the noun is plural. Also, the order of the number and the adjective is reversed, and in some cases, verb agreement rules must be suspended. There are also interesting constraints on the construction itself: for example, some adjectives, such as color words, are not acceptable. Furthermore, qualitative adjectives must appear before quantitative ones. Overall, GPT-3 judgments match the direction of the human ones across a variety of conditions, except on the question of quantitative vs qualitative adjectives, where humans showed no preference, and GPT-3 had a slightly preference against the one described in the literature. This shows that the model understood the syntactic structure of the AANN construction to the point where it can override more global "rules" about word order, but makes no statement about its understanding of the meaning.

4.2. NLP and construction grammar

Other computational studies about CxG have either focused on automatically annotating constructions (Dunietz et al., 2017)

```
S → SPOS | SNEG
SPOS → POS1 PUNCT POS2 '.' | POS1 INSERT PUNCT POS2 '.'
SNEG → NEG1 PUNCT NEG2 '.' | NEG1 INSERT PUNCT NEG2 '.'
PUNCT → ',' | ';' | ϵ
CORE_POS → ADV_I 'the' NUM NOUN VERB
CORE_NEG → ADV_I NUM VERB 'the' NOUN
POS_UPPER → '0 The' CORE_POS
POS_LOWER → '0 the' CORE_POS
NEG_UPPER → '0 The' CORE_NEG
NEG_LOWER → '0 the' CORE_NEG
POS1 → POS_UPPER | POS_UPPER ADD | START POS_LOWER | START POS_LOWER ADD
POS2 → POS_LOWER | POS_LOWER ADD
NEG1 → NEG_UPPER | NEG_UPPER ADD | START NEG_LOWER | START NEG_LOWER ADD
NEG2 → NEG_LOWER | NEG_LOWER ADD
INSERT → INSERT1 | INSERT2
INSERT2 → ADDITION BETWEEN_ADD_AND_SENT SENT
PRON → 'we' | 'they'
ADDITION → ', and by the way,' | ', and I want to add that' | ', and' PRON 'just want to say that' | ', and
then' PRON 'said that' | ', and then' PRON 'said that'
SAY → 'say' | 'think' | 'mean' | 'believe'
BETWEEN_ADD_AND_SENT → PRON SAY 'that' | PRON SAY 'that' | PRON SAY 'that' | PRON SAY 'that'
LOC_SENT → PRON 'said this in' LOC 'too'
LOC → CITY 'and' LOC | CITY
CITY → 'Munich' | 'Washington' | 'Cologne' | 'Prague' | 'Istanbul'
SENT → 'this also holds in other cases' | 'this is not always true' | 'this is always true' | 'this has only
recently been the case' | 'this has not always been the case' | 'this has always been the case'
INSERT1 → 'without stopping' | 'without a break' | 'without a pause' | 'uninterrupted'
START → 'Nowadays, ' | 'Nowadays' | 'Therefore, ' | 'Therefore' | 'We can' CANWORD 'that' | 'It is' KNOWNWORD
'that' | 'It follows that' | 'Sometimes'
START → Sometimes,' | It was recently announced that' | People have told me that' | I recently read in a
really interesting book that' | I have recently read in an established, well-known newspaper that' | It was
reported in a special segment on TV today that'
CANWORD → say' | surmise' | accept' | state'
KNOWNWORD → clear' | known' | accepted' | obvious'
ADD → TEMP | UNDER1 | TEMP UNDER1 | UNDER1 TEMP
ADV_I → ADV | ADV and' ADV
TEMP → TEMP1 TEMP2
TEMP1 → before' | after' | during'
TEMP2 → the morning' | the afternoon' | the night'
UNDER1 → under the' UNDER2
UNDER2 → bed' | roof' | sun'
VERB → push' | attack' | chase' | beat' | believe' | boil' | box' | burn' | call' | date'
NOUN → lions' | pandas' | camels' | pigs' | horses' | sheep' | chickens' | foxes' | cows' | deer'
ADV → worse' | earlier' | slower' | deeper' | bigger' | smaller' | flatter' | weaker' | stronger' | louder'
NUM → twelve' | thirteen' | fourteen' | fifteen' | sixteen' | seventeen' | eighteen' | nineteen' | twenty' |
'twenty-one'
```

Algorithm 1. Context-free grammar for artificial data creation training set.

or on the creation and evaluation of automatically built lists of constructions (Marques and Beuls, 2016; Dunn, 2019).

4.3. General probing

Our work also bears some similarity to recent work in generative grammar-based syntax probing of large PLMs in that we approximate the minimal pairs-based probing framework similar to Wei et al. (2021), Marvin and Linzen (2018), or Goldberg (2019). However, as we are concerned with different phenomena and investigating them from a different theoretical standpoint, the syntactic half of our work clearly differs.

The semantic half of our study is closest to recent work on designing challenging test cases for models such as Ribeiro et al. (2020), who design some edge cases for which most PLMs fail. Despite the different motivation, the outcome is very similar to a list of some particularly challenging constructions.

```
S → SPOS | SNEG
SPOS → POS1 PUNCT POS2 ’.’ | POS1 INSERT PUNCT POS2 ’.’
SNEG → NEG1 PUNCT NEG2 ’.’ | NEG1 INSERT PUNCT NEG2 ’.’
PUNCT → ’,’ | ’;’ | ”
CORE_POS → ADV_I ’the’ NUM NOUN VERB
CORE_NEG → ADV_I NUM VERB ’the’ NOUN
POS_UPPER → ’0 The’ CORE_POS
POS_LOWER → ’0 the’ CORE_POS
NEG_UPPER → ’0 The’ CORE_NEG
NEG_LOWER → ’0 the’ CORE_NEG
POS1 → POS_UPPER | POS_UPPER ADD | START POS_LOWER | START POS_LOWER ADD
POS2 → POS_LOWER | POS_LOWER ADD
NEG1 → NEG_UPPER | NEG_UPPER ADD | START NEG_LOWER | START NEG_LOWER ADD
NEG2 → NEG_LOWER | NEG_LOWER ADD
INSERT → INSERT1 | INSERT2
INSERT2 → ADDITION BETWEEN_ADD_AND_SENT SENT
PRON → ’I’ | ’you’
ADDITION → ’, and by the way ,’ | ’, and I want to add that’ | ’, and’ PRON ’just want to say that’ | ’,
and then’ PRON ’said that’ | ’, and then’ PRON ’said that’
SAY → ’say’ | ’think’ | ’mean’ | ’believe’
BETWEEN_ADD_AND_SENT → PRON SAY ’that’ | PRON SAY ’that’ | PRON SAY ’that’ | PRON SAY ’that’
LOC_SENT → PRON ’said this in’ LOC ’too’
LOC → CITY ’and’ LOC | CITY
CITY → ’London’ | ’New York’ | ’Berlin’ | ’Madrid’ | ’Paris’
SENT → ’this also holds in other cases’ | ’this is not always true’ | ’this is always true’ | ’this has
only recently been the case’ | ’this has not always been the case’ | ’this has always been the case’
INSERT1 → ’without stopping’ | ’without a break’ | ’without a pause’ | ’uninterrupted’ |
START → ’Nowadays ,’ | ’Nowadays’ | ’Therefore ,’ | ’Therefore’ | ’We can’ CANWORD ’that’ | ’It is’
KNOWNWORD ’that’ | ’It follows that’ | ’Sometimes’ | ’Sometimes ,’ | ’It was recently announced that’ |
’People have told me that’ | ’I recently read in a really interesting book that’ | ’I have recently read in
an established , well-known newspaper that’ | ’It was reported in a special segment on TV today that’
CANWORD → ’say’ | ’surmise’
KNOWNWORD → ’clear’ | ’known’
ADD → TEMP | UNDER1 | TEMP UNDER1 | UNDER1 TEMP
ADV_I → ADV | ADV ’and’ ADV
TEMP → TEMP1 TEMP2
TEMP1 → ’before’ | ’after’ | ’during’
TEMP2 → ’the day’ | ’the night’ | ’the evening’
UNDER1 → ’under the’ UNDER2
UNDER2 → ’bridge’ | ’stairs’ | ’tree’
VERB → ’slam’ | ’break’ | ’bleed’ | ’shake’ | ’smash’ | ’throw’ | ’strike’ | ’shoot’ | ’swallow’ | ’choke’
NOUN → ’cats’ | ’dogs’ | ’girls’ | ’boys’ | ’men’ | ’women’ | ’people’ | ’humans’ | ’mice’ | ’alligators’
ADV → ’faster’ | ’quicker’ | ’harder’ | ’higher’ | ’later’ | ’longer’ | ’shorter’ | ’lower’ | ’wider’ |
’better’
NUM → ’two’ | ’three’ | ’four’ | ’five’ | ’six’ | ’seven’ | ’eight’ | ’nine’ | ’ten’ | ’eleven’
```

Algorithm 2. Context-free grammar for artificial data creation test set.

5. Syntax

Our investigation of PLMs' knowledge of the CC is split into two parts. First, we probe for the PLMs' knowledge of the syntactic aspects of the CC, to determine if they recognize its structure. Then we devise a test of their understanding of its semantic aspects by investigating their ability to apply, in a given context, information conveyed by a CC.

5.1. Probing methods

As the first half of our analysis of PLMs' knowledge of the CC, we investigate its syntactic aspects. Translated into probing questions, this means that we ask: can a PLM recognize an instance of the CC? Can it distinguish instances of the CC from similar-looking non-instances? Is it able to go beyond the simple recognition of its fixed parts ("The `COMP-ADJ/ADV`, the ...") and group all ways of completing the sentences that are instances of the CC separately from all those that are not? And to frame all of these questions in a syntactic probing framework: will we be able to recover, using a logistic regression as the probe, this distinguishing information from a PLM's embeddings?

The established way of testing a PLM for its syntactic knowledge has in recent years become minimal pairs (e.g., Warstadt et al., 2020a; Demszky et al., 2021). This would mean pairs of sentences which are indistinguishable except for the fact that one of them is an instance of the CC and the other is not, allowing us to perfectly separate a model's knowledge of the CC from other confounding factors. While this is indeed possible for simpler syntactic phenomena such as verb-noun number agreement, there is no obvious way to construct minimal pairs for the CC. We therefore construct minimal pairs in two ways: one with artificial data based on a context-free grammar (CFG), and one with sentences extracted from C4.

5.1.1. Synthetic data

In order to find a pair of sentences that is as close as possible to a minimal pair, we devise a way to modify the words following "The X-er" such that the sentence is no longer an instance of the construction. The pattern for a positive instance is "The `ADV`-er the `NUM NOUN VERB`," e.g., "The harder the two cats fight." To create a negative instance, we reorder the pattern to "The `ADJ`-er `NUM VERB` the `NOUN`," e.g., "The harder two fight the cats." The change in role of the numeral from the dependent of a head to a head itself, made possible by choosing a verb that can be either transitive or intransitive, as well as the change from an adverb to an adjective, allows us to construct a negative instance that uses the same words as the positive one, but in a different order.[3] In order to generate a large number of instances, we collect two sets each of adverbs, numerals, nouns, and verbs that are mutually exclusive between training and test sets. To investigate if the model is confused by additional content in the sentences, we write an CFG to insert phrases before the start of the first half, in between the two halves, and after the second half of the CC. We show the rules making up the CFG in Algorithms 1, 2.

While this setup is rigorous in the sense that positive and negative sentences are exactly matched, it comes with the drawback of only considering one type of CC. To be able to conduct a more comprehensive investigation, we adopt a complementary approach and turn to pairs extracted from C4. We show examples of training

[3] Note that an alternative reading of this sentence exists: the numeral "two" forms the noun phrase by itself and "The harder" is still interpreted as part of the CC. The sentence is actually a positive instance on this interpretation. We regard this reading as very improbable.

TABLE 3 Examples of data for the syntactic probe.

Sentence	Label	Source
"The higher up the nicer!"	Positive	Corpus
She thinks the more water she drinks the better her skin looks.	Positive	Corpus
Subtract the smaller from the larger.	Negative	Corpus
The way the older guys help out the younger guys is fantastic.	Negative	Corpus
Nowadays, the bigger the 18 sheep date, the louder and bigger the 12 horses beat under the sun.	Positive	Artificial train
The flatter the 14 lions push, the deeper and smaller the 16 deer burn under the roof.	Positive	Artificial train
Sometimes, the worse and earlier 17 believe the deer, and we just want to say that they mean that this has always been the case, the flatter 21 attack the foxes before the afternoon under the roof.	Negative	Artificial train
Nowadays, the smaller 16 box the camels, and by the way, they mean that this is always true; the weaker 13 date the cows.	Negative	Artificial train
The harder and longer the three cats throw, the harder and shorter the 10 dogs shake.	Positive	Artificial test
I have recently read in an established, well-known newspaper that the later the ten mice strike; the later and better the seven men smash under the tree during the night.	Positive	Artificial test
The higher nine strike the women without a pause the shorter 10 choke the girls.	Negative	Artificial test
We can say that the longer and faster four strike the men under the stairs before the evening, the harder four throw the dogs after the day under the bridge.	Negative	Artificial test

and test data in Table 3. These cover a broad range of CC patterns, albeit without meeting the criterion that positive and negative samples are exactly matched.

5.1.2. Corpus-based minimal pairs

While accepting that positive and negative instances extracted from a corpus will automatically not be minimal and therefore contain some lexical overlap and context cues, we attempt to regularize our retrieved instances as far as possible. To form a first candidate set, we POS tag C4 using spaCy (Honnibal and Montani, 2018) and extract all sentences that follow the pattern "The" (DET) followed by either "more" and an adjective or adverb, or an adjective or adverb ending in "-er," and at any point later in the sentence again the same pattern. We discard examples with adverbs or adjectives that were falsely labeled as comparative, such as "other." We then group these sentences by their sequence of POS tags, and manually classify the sequences as either positive or negative instances. We observe that sentences sharing a POS tag pattern tend to be either all negative or all positive instances, allowing us to save annotation time by working at the POS tag pattern level instead of the sentence level. To make the final set as diverse as possible, we sort the patterns randomly and label as many as possible. In order to further reduce interfering factors in our probe, we separate the POS tag patterns between training and test sets. We give examples in Table 3.

Please note that due to the inherent difficulty of creating minimal pairs for this construction, while the two approaches are complementary, neither of them is perfect. While we think that our experimental setup (e.g., no surface patterns indicating positive/negative classes, clear distinction between training/test data) is designed well-enough, we would like to note that probing classifiers with logistic regression are not robust to such confound variables.

5.1.3. The probe

For both datasets, we investigate the overall accuracy of our probe as well as the impact of several factors. The probe consists of training a simple logistic regression model on top of the mean-pooled sentence embeddings (Vulić et al., 2020). To quantify the impact of the length of the sentence, the start position of the construction, the position of its second half, and the distance between them, we construct four different subsets D_f^{train} and D_f^{test} from both the artificially constructed and the corpus-based dataset. For each subset, we sample sentences such that both the positive and the negative class is balanced across every value of the feature within a certain range of values. This ensures that the probes are unable to exploit correlations between a class and any of the above features. We create the dataset as follows

$$D_f = \bigcup_{v \in f_v} \bigcup_{l^* \in L} S(D, v, l^*, n^*),$$

where f is the feature, f_v is the set of values for f, $L = \{positive, negative\}$ are the labels, and S is a function that returns n^* elements from D that have value v and label l^*.

To make this task more cognitively realistic, we aim to test if a model is able to generalize from shorter sentences, which contain relatively little additional information besides the parts relevant to the classification task, to those with greater potential interference due to more additional content that is not useful for classification. Thus, we restrict the training set to samples from the lowest quartile of each feature so that f_v becomes $[v_f^{\min}, v_f^{\min} + \frac{1}{4}(v_f^{\max} - v_f^{\min})]$ for D_f^{train} and $[v_f^{\min}, v_f^{\max}]$ for D_f^{test}. We report the test performance for every value of a given feature separately to recognize patterns. For the artificial syntax probing, we generate 1,000 data points for each value of each feature for each training and test for each subset associated with a feature. For the corpus syntax probing, we collect 9,710 positive and 533 negative sentences in total, from which we choose 10 training and five test sentences for each value of each feature in a similar manner. To improve comparability

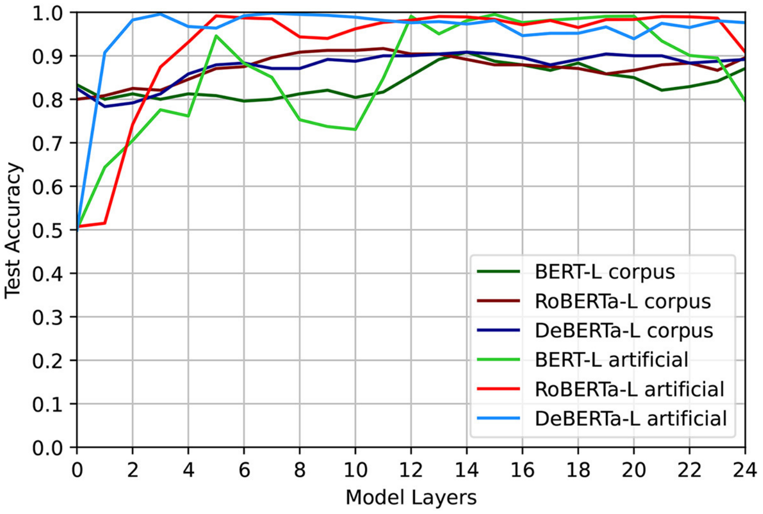


FIGURE 1
Overall accuracy per layer for D_{length}. All shown models are the large model variants. The models can easily distinguish between positive and negative examples in at least some of their layers.

and make the experiment computationally feasible, we test the "large" size of each of our three models, using the Huggingface Transformers library (Wolf et al., 2019). Our logistic regression probes are implemented using Scikitlearn (Pedregosa et al., 2011).

5.2. Probing results

5.2.1. Artificial data

As shown in Figure 1, the results of our syntactic probe indicate that all models can easily distinguish between positive and negative examples in at least some of their layers, independently of any of the sentence properties that we have investigated. We report full results in Figures A1–A3 in the Appendix (Supplementary material). We find a clear trend that DeBERTa performs better than RoBERTa, which in turn performs better than BERT across the board. As DeBERTa's performance in all layers is nearly perfect, we are unable to observe patterns related to the length of the sentence, the start position of the CC, the start position of the second half of the CC, and the distance between them. By contrast, we observe interesting patterns for BERT and RoBERTa. For D_{length}, and to a lesser degree D_{distance} (which correlates with it), we observe that at first, performance goes down with increased length as we would expect—the model struggles to generalize to longer sentences with more interference since it was only trained on short ones. However, this trend is reversed in the last few layers. We hypothesize this may be due to an increased focus on semantics in the last layers (Peters et al., 2018; Tenney et al., 2019), which could lead to interfering features particularly in shorter sentences.

5.2.2. Corpus data

In contrast, the results of our probe on more natural data from C4 indicate two different trends: first, as the positive and negative instances are not identical on a bag-of-word level, performance is not uniformly at 50% (i.e., chance) level in the first layers, indicating that the model can exploit lexical cues to some degree. We observe a similar trend as with the artificial experiment, which showed that DeBERTa performs best and BERT worst. The corresponding graphs can be found in Figures A4–A6 in Supplementary material.

Generally, this additional corpus-based experiment validates our findings from the experiment with artificially generated data, as all models perform at 80% or better from the middle layers on, indicating that the models are able to classify instances of the construction even when they are very diverse and use unseen POS tag patterns.

Comparing the average accuracies on D_{length} for both data sources in Figure 1, we observe that all models perform better on artificial than on corpus data from the fifth layer on, with the notable exception of a dip in performance for BERT large around layer 10.

6. Semantics

6.1. Probing approach

For the second half of our investigation, we turn to semantics. In order to determine if a model has understood the meaning of the CC, i.e., if it has understood that in any sentence, "the `COMP` .... the `COMP`" implies a correlation between the two halves, we adopt a usage-based approach and ask: can the model, based on the meaning conveyed by the CC, draw a correct inference in a specific scenario? For this, we construct general test instances of the CC that consist of a desired update of the belief state of the model about the world, which we then expect it to be able to apply. More concretely, we generate sentences of the form "The `ADJ1`-er you are, the `ADJ2`-er you are.," while picking adjectives at random. To this general statement, we then add a specific scenario with two random names: "`NAME1` is `ADJ1`-er than `NAME2`." and ask the model to draw an inference from it. We first construct a test scenario for this that works with masked language models and test BERT, RoBERTa and DeBERTa on it, and then modify the setup and move on to autoregressive models, specifically OPT (Zhang et al., 2022).

6.2. Experiments on masked language models

6.2.1. Probing methods

In our experiments with masked language models, we now ask the models to draw an inference from the context by predicting a token at the masked position in the following sentence: "Therefore, `NAME1` is `[MASK]` than `NAME2`." If the model has understood the

TABLE 4 Overview of constructions investigated in CxG-specific probing literature, with examples.

References	Language	Source	Construction	Example
Tayyar Madabushi et al. (2020)	English	From automatically constructed list by Dunn (2017)	Personal Pronoun + didn't + V + how	We didn't know how or why.
Li et al. (2022)	English	Argument structure constructions according to Bencini and Goldberg (2000)	caused-motion	Bob cut the bread into the pan.
Tseng et al. (2022)	Chinese	From constructions list by Zhan (2017)	a + 到+ 爆, etc.	好吃到爆了！ *It's so delicious!*
Weissweiler et al. (2022)	English	McCawley (1988)	Comparative correlative	The bigger, the better.
Mahowald (2023)	English	Jackendoff (1977)	Article + Adjective + Numeral + Noun	A lovely 5 days

TABLE 5 Overview of the schemata of all test scenarios used for semantic probing for masked language models.

No.	Purpose	Approach	Sentence schema
S1	Base		The ADJ1-er you are, the ADJ2-er you are. The ANT1-er you are, the ANT2-er you are. NAME1 is ADJ1-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.
S2	Bias test	Recency	The ANT1-er you are, the ANT2-er you are. The ADJ1-er you are, the ADJ2-er you are. NAME1 is ADJ1-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.
S3		Vocabulary	The ADJ1-er you are, the ANT2-er you are. The ANT1-er you are, the ADJ2-er you are. NAME2 is ADJ1-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.
S4		Name	The ADJ1-er you are, the ADJ2-er you are. The ANT1-er you are, the ANT2-er you are. NAME2 is ADJ1-er than NAME1. Therefore, NAME2 is [MASK] than NAME1.
S5	Calibration	Short	NAME1 is ADJ1-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.
S6		Name	The ADJ1-er you are, the ADJ2-er you are. The ANT1-er you are, the ANT2-er you are. NAME1 is ADJ1-er than NAME2. Therefore, NAME3 is [MASK] than NAME4.
S7		Adjective	The ADJ1-er you are, the ADJ2-er you are. The ANT1-er you are, the ANT2-er you are. NAME1 is ADJ3-er than NAME2. Therefore, NAME1 is [MASK] than NAME2.

meaning conveyed by the CC and is able to use it in predicting the mask, we expect the probability of ADJ2 to be high.

To provide the model with an alternative, we add a second sentence, another instance of the CC, using the antonyms of the two adjectives. This sentence is carefully chosen to have no impact on the best filler for [MASK], but also for other reasons explained in Section 6.2.1.1. The full test context is shown in Table 5, S1. This enables us to compare the probability of ADJ2 for the mask token directly with a plausible alternative, ANT2. One of our test sentences might be "The stronger you are, the faster you are. The weaker you are, the slower you are. Terry is stronger than John. Therefore, Terry will be [MASK] than John," where we compare the probabilities of "faster" and "slower."

Note that success in our experiment does not necessarily indicate that the model has fully understood the meaning of the CC. The experiment can only provide a lower bound for the underlying understanding of any model. However, we believe that our task is not unreasonable for a masked language model in a zero-shot setting. It is comparable in difficulty and non-reliance on world knowledge to the NLU tasks presented in LAMBADA (Paperno et al., 2016), on which GPT-2 (117 M to 1.5 B parameters) has achieved high zero-shot accuracy (Radford et al., 2019, Table 4). While we investigate masked language models and not GPT-2, our largest models are comparable in size to the sizes of GPT-2 that were used (340 M for BERT$_{\text{L}}$, 355 M for RoBERTa$_{\text{L}}$, and 1.5 B parameters for DeBERTa-XXL$_{\text{L}}$), and we believe that this part of our task is achievable to some degree.

6.2.1.1. Biases

In this setup, we hypothesize several biases that models could exhibit and might cloud our assessment of its understanding of the CC, and devise a way to test their impact.

Firstly, we expect that models might prefer to repeat the adjective that is closest to the mask token. This has recently been documented for prompt-based experiments (Zhao et al., 2021). Here, this adjective is ANT2, the wrong answer. To test the influence this has on the prediction probabilities, we construct an alternative version of our test context in which we flip the first two sentences so that the correct answer is now more recent. The result can be found in Table 5, S2.

Secondly, we expect that models might assign higher probabilities to some adjectives, purely based on their frequency in the pretraining corpus, as for example observed by Holtzman et al. (2021). To test this, we construct a version of the test context in which ADJ2/ANT2 are swapped, which means that we can keep both the overall words the same as well as the position of the correct answer, while changing which adjective it is. The sentence is now S3 in Table 5. If there is a large difference between the prediction probabilities for the two different versions, that this means that a model's prediction is influenced by the lexical identity of the adjective in question.

Lastly, a model might have learned to associate adjectives with names in pretraining, so we construct a third version, in which we swap the names. This is S4 in Table 5. If any prior association between names and adjectives influences the prediction, we expect the scores between S4 and S1 to differ.

6.2.1.2. Calibration

After quantifying the biases that may prevent us from seeing a model's true capability in understanding the CC, we aim to develop methods to mitigate it. We turn to calibration, which has recently been used in probing with few-shot examples by Zhao et al. (2021). The aim of calibration is to improve the performance of a model on a classification task, by first assessing the prior probability of a label (i.e., its probability if no context is given), and then dividing the probability predicted in the task context by this prior; this gives us the conditional probability of a label given the context, representing the true knowledge of the model about this task. In adapting calibration, we want to give a model every possible opportunity to do well so that we do not underestimate its underlying comprehension.

We therefore develop three different methods of removing the important information from the context in such a way that we can use the prediction probabilities of the two adjectives in these

TABLE 6 Selected accuracies and results for the semantic probe.

	Accuracy		Decision flip		
	S1	S2	S2	S3	S4
BERT$_{\text{B}}$	37.65	64.64	26.98	75.69	02.70
BERT$_{\text{L}}$	36.85	67.21	30.44	73.31	02.32
RoBERTa$_{\text{B}}$	61.60	52.84	09.91	76.18	02.76
RoBERTa$_{\text{L}}$	55.71	68.00	14.33	79.47	04.33
DeBERTa$_{\text{B}}$	49.72	49.80	00.91	99.66	01.07
DeBERTa$_{\text{L}}$	50.88	51.40	07.04	94.83	02.23
DeBERTa$_{\text{XL}}$	47.73	49.33	05.46	89.28	02.51
DeBERTa$_{\text{XXL}}$	47.34	48.72	03.59	82.09	01.13

We report the average accuracy on the more difficult sentences in terms of recency bias (S1) and the easier ones (S2), as well as the percentage of decisions flipped by changing from the base S1 to the sentences testing for recency bias (S2), vocabulary bias (S3), and name bias (S4). RoBERTa and DeBERTa perform close to chance on S1 and S2 accuracy, indicating that they do not understand the meaning of CC. BERT's performance is strongly influenced by biases (recency, lexical identity), also indicating that it has very limited if any understanding of CC.

contexts for calibration. The simplest way of doing this is to remove both instances of the CC, resulting in S5 in Table 5. If we want to keep the CC in the context, the two options to remove any information are to replace either the names or the adjectives with new names/adjectives. We therefore construct two more instances for calibration: S6 and S7 in Table 5.

For each calibration method, we collect five examples with different adjectives or names. For a given base sample S_b, we calculate P_c, the calibrated predictions, as follows:

$$P_c(a|S_b) = P(a|S_b)/[\sum_{i=1}^{i=5}(P(a|C_i)/5)]$$

where C_i is the i-th example of a given calibration technique, a is the list of adjectives tested for the masked position, and the division is applied elementwise. We collect a list of 20 adjectives and their antonyms manually from the vocabulary of the RoBERTa tokenizer and 33 common names and generate 144,800 sentences from them. We test BERT (Devlin et al., 2019) in the sizes base and large, RoBERTa (Liu et al., 2019) in the sizes base and large, and DeBERTa (He et al., 2020) in the sizes base, large, xlarge, and xxlarge.

6.2.2. Results

In Table 6, we report the accuracy for all examined models. Out of the three variations to test biases, we report accuracy only for the sentence testing the recency bias as we expect this bias to occur systematically across all sentences: if it is a large effect, it will always lead to the sentence where the correct answer is the more recent one being favored. To assess the influence of each bias beyond accuracy, we report as decision flip the percentage of sentences for which the decision (i.e., if the correct adjective had a higher probability than the incorrect one) was changed when considering the alternative sentence that was constructed to test for bias. We report full results in Table 7.

Looking at the accuracies, we see that RoBERTa's and DeBERTa's scores are close to 50 (i.e., chance) accuracy for both S1 and S2. BERT models differ considerably as they seem to suffer from bias related to the order of the two CCs, but we can see that the average between them is also very close to chance. When we further look at the decision flips for each of the biases, we find that there is next to no bias related to the choice of names (S4). However, we can see a large bias related to both the recency of the correct answer (S2) and the choice of adjectives (S3). The recency bias is strongest in the BERT models, which also accounts for the difference in accuracies. For RoBERTa and DeBERTa models, the recency bias is small, but clearly present. In contrast, they exhibit far greater bias toward the choice of adjective, even going as far as 99.66% of decisions flipped by changing the adjective for DeBERTa base. This suggests that these models' decisions about which adjective to assign a higher probability is almost completely influenced by the choice of adjective, not the presence of the CC. Overall, we conclude that without calibration, all models seem to be highly susceptible to different combinations of bias, which completely obfuscate any underlying knowledge of the CC, leading to an accuracy at chance level across the board.

We therefore turn to our calibration methods, evaluating them first on their influence on the decision flip scores, which directly show if we were able to reduce the impact of the different types of bias. We report these only for order and vocabulary bias as we found name bias to be inconsequential. We report the complete results in Table 7. We see that across all models, while all three calibration methods work to reduce some bias, none does so consistently across all models or types of bias. Even in cases where calibration has clearly reduced the decision flip score, we find that the final calibrated accuracy is still close to 50%. This indicates that despite the effort to retrieve any knowledge that the models have about the CC, they are unable to perform clearly above chance, and we have therefore found no evidence that the investigated models understand and can use the semantics of the CC.

To investigate if this was result was exclusive to smaller, masked language models, we repeat our experiment and turn to larger, autoregressive models, more specifically, different sizes of OPT (Zhang et al., 2022).

6.3. Experiments on autoregressive language models

6.3.1. Methods

6.3.1.1. Probing setup

Since we concluded from our experiments with masked language models that none of them have reached significant performance on our task, we move on to investigating newer autoregressive models. We hope that as these models have been shown to perform significantly better on natural language understanding (NLU; Zhang et al., 2022), which is a prerequisite for our probing setup, their performance will be more directly indicative of their understanding of the CC in context.

As we can no longer perform our experiments on the basis of comparing the predictions for a given MASK token, we modify the setup such that our metric is based on the comparison of the perplexity of two competing whole sentences. Our main idea is to no longer work with antonyms but instead create contrast

TABLE 7 Accuracies for the semantic probe with our three calibration methods compared to no calibration.

Model	Test sentence	Accuracies				Decision flips			
		–	S5	S6	S7	–	S5	S6	S7
BERT$_B$	S1	37.65	37.62	44.39	47.9	–	–	–	–
	S2	64.64	62.79	56.66	55.41	26.99	25.22	14.75	10.77
	S3	38.04	44.78	44.09	48.29	75.69	23.51	86.33	91.05
	S4	–	–	–	–	2.71	–	–	–
BERT$_L$	S1	36.85	31.91	47.21	44.03	–	–	–	–
	S2	67.13	73.48	54.39	64.45	30.44	41.8	13.37	22.24
	S3	36.46	43.43	47.79	44.36	73.31	25.94	88.65	85.97
	S4	–	–	–	–	2.32	–	–	–
RoBERTa$_B$	S1	61.6	58.76	42.13	62.32	–	–	–	–
	S2	52.85	51.35	71.33	60.25	9.92	8.67	31.13	10.86
	S3	62.21	55.17	43.04	62.76	76.19	22.04	79.03	74.75
	S4	–	–	–	–	2.76	–	–	–
RoBERTa$_L$	S1	55.72	58.37	65.08	69.53	–	–	–	–
	S2	68.01	74.53	62.73	77.76	14.34	17.82	15.94	15.86
	S3	55.36	52.02	65.28	69.23	79.48	43.64	79.75	78.32
	S4	–	–	–	–	3.25	–	–	–
DeBERTa$_B$	S1	41.61	36.41	32.79	43.27	–	–	–	–
	S2	42.95	43.04	33.77	42.36	24.21	24.4	8.79	7.49
	S3	41.92	38.64	32.39	43.31	74.58	17.83	72.29	64.42
	S4	–	–	–	–	1.67	–	–	–
DeBERTa$_L$	S1	58.5	60.34	45.17	65.42	–	–	–	–
	S2	64.56	66.43	49.99	62.77	13.47	14.27	14.43	13.15
	S3	58.8	59.84	45.41	65.45	78.25	30.36	75.61	70.21
	S4	–	–	–	–	2.65	–	–	–
DeBERTa$_{XL}$	S1	67.24	74.59	57.33	76.64	–	–	–	–
	S2	76.31	78.92	63.75	78.41	18.02	18.79	17.37	16.48
	S3	67.28	74.35	57.51	76.69	82.35	43.29	78.43	72.99
	S4	–	–	–	–	3.34	–	–	–

We report the average accuracy on the more difficult sentences in terms of recency bias (S1), the easier ones (S2), and vocabulary bias (S3), as well as the percentage of decisions flipped by changing from the base S1 to each sentence. Our calibration techniques are short (S5), name (S6), and adjective (S7).

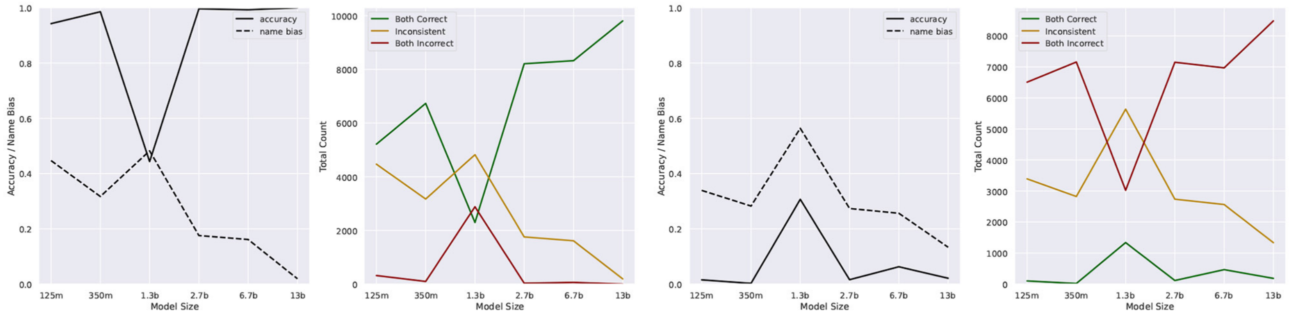


FIGURE 2
Accuracy and name bias scores for test sentences S8–S11 on the left and S12–S15 on the right, on different sizes of OPT.

TABLE 8 Overview of the schemata of test scenarios S8–S15, used for semantic probing for autoregressive language models.

No.	Name order	Validity	Sentence schema
S8	Same	True	The ADJ1-er you are, the ADJ2-er you are. NAME1 is ADJ1-er than NAME2. Therefore, NAME1 will be ADJ2-er than NAME2.
S9		False	The ADJ1-er you are, the ADJ2-er you are. NAME1 is ADJ1-er than NAME2. Therefore, NAME2 will be ADJ2-er than NAME1.
S10		True	The ADJ1-er you are, the ADJ2-er you are. NAME2 is ADJ1-er than NAME1. Therefore, NAME2 will be ADJ2-er than NAME1.
S11		False	The ADJ1-er you are, the ADJ2-er you are. NAME2 is ADJ1-er than NAME1. Therefore, NAME1 will be ADJ2-er than NAME2.
S12	Flipped	True	The ADJ1-er you are, the ADJ2-er you are. NAME1 is less ADJ1 than NAME2. Therefore, NAME2 will be ADJ2-er than NAME1.
S13		False	The ADJ1-er you are, the ADJ2-er you are. NAME1 is less ADJ1 than NAME2. Therefore, NAME1 will be ADJ2-er than NAME2.
S14		True	The ADJ1-er you are, the ADJ2-er you are. NAME2 is less ADJ1 than NAME1. Therefore, NAME1 will be ADJ2-er than NAME2.
S15		False	The ADJ1-er you are, the ADJ2-er you are. NAME2 is less ADJ1 than NAME1. Therefore, NAME2 will be ADJ2-er than NAME1.

by swapping the two names in the last sentence. Given the context "The ADJ1-er you are, the ADJ2-er you are. NAME1 is ADJ1-er than NAME2.," we contrast the perplexities of "Therefore, NAME1 will be ADJ2-er than NAME2" and "Therefore, NAME2 will be ADJ2-er than NAME1." While the sentences are bag-of-words equivalent, only the first one follows from the context. This has the additional effect of removing the confounding factor of the second sentence with antonyms from the factors that influence the model's performance. For example, we would now contrast "The stronger you are, the faster you are. Terry is stronger than John. Therefore, Terry will be faster than John" with "The stronger you are, the faster you are. Terry is stronger than John. Therefore, John will be faster than Terry."

6.3.1.2. Name bias

Similarly to our previous experiment in Section 6.2.1, we hypothesize biases to this setup and test them. Our "adjective bias" and "recency bias" are not immediately applicable here, as we no longer have a masked token.

However, we expect that models might consistently prefer one final sentence, which is the one that changes the acceptability of the entire test phrase, over another, regardless of context. To test this, we construct a second pair of sentences, where the names are swapped both times. This means that when iterating through all 4-tuples of sentences that belong together, we can now compare all four and count only those as valid results where either both pairs were correctly classified or both were incorrectly classified. For the others, where one was correct and the other incorrect, this indicates that the model preferred one final sentence over the other in all contexts. We count how many times this occurs to quantify the strength of this name bias in a model.

6.3.2. Initial results

For our results, we consider each four-tuple of sentences S_8-S_{11}. We perform perplexity comparisons twice: firstly, we expect the perplexity of S_8 to be lower than that of S_9; secondly, we anticipate the perplexity of S_{10} to be lower than that of S_{11}. We denote C to represent the count of correct results where both conditions are met, I to represent the count of incorrect results where both conditions fail, and In to represent the count of inconclusive results where one condition is met and the other is not.

The general trend for these three counts can be seen in the right half of Figure 2. As the models increase in size, C rises and In drops, with I remaining generally low. The only exception to this is the OPT-1.3b model, for unknown reasons.

We then develop two more abstract metrics based on these counts:

1. We define the accuracy, A, as the number of correct responses divided by the number of valid responses (correct and incorrect ones). In mathematical terms: $A = \frac{C}{C+I}$.
2. As a complementary metric, we define the "name bias," B, as the percentage of inconclusive responses over total responses. Mathematically, $B = \frac{In}{C+I+In}$.

We use "name bias" to denote situations where the model consistently favored one of the two possibilities for the last sentence, indicating a possible bias for this sentence, perhaps due to the order of names and the combination with the particular adjective.

Our observations show that A remains consistently high (with the exception of 1.3 b) and B decreases as the model size increases.

These results were initially encouraging for the hypothesis that larger, autoregressive models are able to capture the semantics of the CC. However, there is one important possibility for bias in all four sentences: the correct answer is consistently that in which the two names are in the same order in both sentences. We therefore have to examine the possibility that the near-perfect accuracy displayed in our task is merely due to the name order being parallel and not to any deeper understanding of the sentences.

6.3.3. Additional experiment

We therefore construct four additional sentences, named S12–S15 in Table 8. They are constructed with "less," to ensure that the correct answer is now the one where the order of names is swapped. We rerun the same experiment as before with these sentences. We expect that if the model was merely preferring the parallel order of names, the accuracy would be close to zero, whereas a continued good accuracy would indicate that it formed a deeper understanding of the task.

The results in Figure 2 show that unfortunately the former was the case: all values are approximately inverted compared to the first experiment. If the model had formed an understanding of the CC in this task, our reformulation of the task could not have completely destroyed the performance. We therefore conclude that none of the models, at least in this setup, have demonstrated an understanding of the CC.

6.4. Problem analysis

Different conclusions might be drawn as to why none of these models have learned the semantics of the CC. Firstly, they might not have seen enough examples of it to have formed a general understanding. Given the amount of examples that we were able to find in C4, and the overall positive results from the syntax section, we find this to be unlikely. Secondly, it could be argued that models have never had a chance to learn what the CC means because they have never seen it together with a context in which it was immediately applied, and do not have the same opportunities as humans available, which would be to either interact with the speaker to clarify the meaning, or to make deductions using observations in the real world. This is in line with other considerations about large PLMs acquiring advanced semantics, even though it has for many phenomena been shown that pre-training is enough (Radford et al., 2019). Lastly, it might be possible that the type of meaning representation required to solve this task is beyond the current transformer-style architectures. Overall, our finding that PLMs do not learn the semantics of the CC adds to the growing body of evidence that complex semantics like negation (Kassner and Schütze, 2020) is still beyond state-of-the-art PLMs.

7. Conclusion

We have made a first step toward a thorough investigation of the compatibility of the paradigm of CxG and the syntactic and semantic capabilities exhibited by state-of-the-art large PLMs. For this, we chose the English comparative correlative, one of the most well-studied constructions, and investigated if large PLMs based on masked language modeling have learned it, both syntactically and semantically. We found that even though they are able to classify sentences as instances of the construction even in difficult circumstances, they do not seem to be able to extract the meaning it conveys and use it in context, indicating that while the syntactic aspect of the CC is captured in pretraining of these models, the semantic aspect is not. We then repeated a modified version of our semantic experiments with larger, autoregressive language models, and found that they were similarly unable to capture the semantics of the construction.

8. Limitations

As our experimental setup requires significant customization with regards to the properties of the specific construction we investigate, we are unable to consider other constructions or other languages in this work. We hope to be able to extend our experiments in this direction in the future. Our analysis is also limited—as all probing papers are—by the necessary indirectness of the probing tasks: we cannot directly assess the model's internal representation of the CC, but only construct tasks that might show it but are imperfect and potentially affected by external factors.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found at: https://github.com/LeonieWeissweiler/ComparativeCorrelative.

Author contributions

LW designed the original research question together with VH. They wrote and discussed the linguistics background as well as the motivation for the experiments and the specifics of the syntax experiments. The specifics of the semantics experiments were designed by LW and AK. HS acted as main advisor to LW, VH, and AK and gave guidance on the process throughout. All authors contributed to the article and approved the submitted version.

Funding

This work was funded by the European Research Council (#740516). VH was also supported by the German Academic Scholarship Foundation. AK was also supported by the German Federal Ministry of Education and Research (BMBF, Grant No. 01IS18036A).

Acknowledgments

The content of this manuscript has been presented in part at the 2022 Conference on Empirical Methods in Natural Language Processing (Weissweiler et al., 2022), and the Construction Grammars and NLP (CxGs+NLP) Workshop at the Georgetown University Round Table on Linguistics (GURT; Weissweiler et al., 2023). We are very grateful to David Mortensen and Lori Levin for helpful discussions and comments.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2023.1225791/full#supplementary-material

References

Abeillé, A., and Borsley, R. D. (2008). Comparative correlatives and parameters. *Lingua* 118, 1139–1157. doi: 10.1016/j.lingua.2008.02.001

Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., and Glass, J. (2017). "What do neural machine translation models learn about morphology?," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vancouver, BC: Association for Computational Linguistics), 861–872. doi: 10.18653/v1/P17-1080

Bencini, G. M., and Goldberg, A. E. (2000). The contribution of argument structure constructions to sentence meaning. *J. Mem. Lang.* 43, 640–651. doi: 10.1006/jmla.2000.2757

Bloomfield, L. (1933). *Language*. New York, NY: Holt, Rinehart & Winston.

Chomsky, N. (1988). Generative grammar. Studies in English linguistics and literature.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). "What you can cram into a single $&!#* vector: probing sentence embeddings for linguistic properties," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Melbourne, VIC: Association for Computational Linguistics), 2126–2136. doi: 10.18653/v1/P18-1198

Culicover, P. W., and Jackendoff, R. (1999). The view from the periphery: the English comparative correlative. *Linguist. Inq.* 30, 543–571. doi: 10.1162/002438999554200

Demszky, D., Sharma, D., Clark, J. H., Prabhakaran, V., and Eisenstein, J. (2021). "Learning to recognize dialect features," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HTL) 2021*. doi: 10.18653/v1/2021.naacl-main.184

Den Dikken, M. (2005). Comparative correlatives comparatively. *Linguist. Inq.* 36, 497–532. doi: 10.1162/002438905774464377

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, MI: Association for Computational Linguistics), 4171–4186.

Dunietz, J., Levin, L., and Carbonell, J. (2017). Automatically tagging constructions of causation and their slot-fillers. *Trans. Assoc. Comput. Linguist.* 5, 117–133. doi: 10.1162/tacl_a_00050

Dunn, J. (2017). Computational learning of construction grammars. *Lang. Cogn.* 9, 254–292. doi: 10.1017/langcog.2016.7

Dunn, J. (2019). "Frequency vs. association for constraint selection in usage-based construction grammar," in *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics* (Minneapolis, MI: Association for Computational Linguistics), 117–128. doi: 10.18653/v1/W19-2913

Fillmore, C. J. (1986). "Varieties of conditional sentences," in *Eastern States Conference on Linguistics, Vol. 3*, 163–182. Available online at: https://books.google.de/books/about/Proceedings_of_the_Eastern_States_Confer.html?id=QQiNZntjqRYC&redir_esc=y

Fillmore, C. J. (1989). "Grammatical construction: theory and the familiar dichotomies," in *Language Processing in Social Context*, eds R. Dietrich and C. F. Graumann (Amsterdam: North-Holland), 17–38. doi: 10.1016/B978-0-444-87144-2.50004-5

Fillmore, C. J., Kay, P., and O'Connor, M. C. (1988). Regularity and idiomaticity in grammatical constructions: the case of let alone. *Language* 64, 501–538. doi: 10.2307/414531

Goldberg, A. (2006). *Constructions at Work: The Nature of Generalization in Language*. Oxford, UK: Oxford University Press. doi: 10.1093/acprof:oso/9780199268511.001.0001

Goldberg, A. (1995). *Constructions: A Construction Grammar Approach to Argument Structure*. Chicago, IL; London: University of Chicago Press.

Goldberg, A. E. (2003). Constructions: a new theoretical approach to language. *Trends Cogn. Sci.* 7, 219–224. doi: 10.1016/S1364-6613(03)00080-9

Goldberg, A. E. (2013). "Chapter: 1415 Constructionist approaches," in *The Oxford Handbook of Construction Grammar*, eds T. Hoffmann, and G. Trousdale (Oxford: Oxford University Press), 1531. doi: 10.1093/oxfordhb/9780195396683.013.0002

Goldberg, Y. (2019). Assessing Bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Goldberg, A. E. (1995). *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press.

He, P., Liu, X., Gao, J., and Chen, W. (2020). "Deberta: decoding-enhanced Bert with disentangled attention," in *International Conference on Learning Representations*.

Hilpert, M. (2006). A synchronic perspective on the grammaticalization of Swedish future constructions. *Nordic J. Linguist.* 29, 151–173. doi: 10.1017/S0332586506001569

Hoffmann, T., Horsch, J., and Brunner, T. (2019). The more data, the better: a usage-based account of the english comparative correlative construction. *Cogn. Linguist.* 30, 1–36. doi: 10.1515/cog-2018-0036

Hoffmann, T., and Trousdale, G. (2013). *The Oxford Handbook of Construction Grammar.* Oxford: Oxford University Press. doi: 10.1093/oxfordhb/9780195396683.001.0001

Holtzman, A., West, P., Shwartz, V., Choi, Y., and Zettlemoyer, L. (2021). "Surface form competition: why the highest probability answer isn't always right," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Punta Cana: Association for Computational Linguistics), 7038–7051. doi: 10.18653/v1/2021.emnlp-main.564

Honnibal, M., and Montani, I. (2018). spaCy 2: natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Iwasaki, E., and Radford, A. (2009). "Comparative correlatives in English: a minimalist-cartographic analysis," in *Essex Research Reports in Linguistics*, Vol. 57 (Essex).

Jackendoff, R. (1977). *X Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT Press.

Kassner, N., and Schütze, H. (2020). "Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics), 7811–7818. doi: 10.18653/v1/2020.acl-main.698

Kay, P., and Fillmore, C. J. (1999). Grammatical constructions and linguistic generalizations: the What's X doing Y? Construction. *Language* 75, 1–33. doi: 10.2307/417472

Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind.* Chicago, IL: University of Chicago Press. doi: 10.7208/chicago/9780226471013.001.0001

Langacker, R. W. (1987). *Foundations of Cognitive Grammar: Theoretical Prerequisites*. Stanford, CA: Stanford University Press.

Li, B., Zhu, Z., Thomas, G., Rudzicz, F., and Xu, Y. (2022). "Neural reality of argument structure constructions," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Dublin: Association for Computational Linguistics), 7410–7423. doi: 10.18653/v1/2022.acl-long.512

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: a robustly optimized Bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mahowald, K. (2023). A discerning several thousand judgments: GPT-3 rates the article+ adjective+ numeral+ noun construction. *arXiv preprint: arXiv:2301.12564.* doi: 10.18653/v1/2023.eacl-main.20

Marques, T., and Beuls, K. (2016). "Evaluation strategies for computational construction grammars," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (Osaka: The COLING 2016 Organizing Committee), 1137–1146.

Marvin, R., and Linzen, T. (2018). "Targeted syntactic evaluation of language models," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels: Association for Computational Linguistics), 1192–1202. doi: 10.18653/v1/D18-1151

McCawley, J. D. (1988). "The comparative conditional construction in English, German, and Chinese," in *Annual Meeting of the Berkeley Linguistics Society*, Vol. 14 (Berkeley, CA), 176–187. doi: 10.3765/bls.v14i0.1791

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, N. Q., Bernardi, R., Pezzelle, S., et al. (2016). "The LAMBADA dataset: Word prediction requiring a broad discourse context," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Berlin: Association for Computational Linguistics), 1525–1534. doi: 10.18653/v1/P16-1144

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). "Deep contextualized word representations," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2018* (New Orleans, LO). doi: 10.18653/v1/N18-1202

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog* 1, 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res*. 21, 1–67.

Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Coenen, A., Pearce, A., et al. (2019). "Visualizing and measuring the geometry of Bert," in *Advances in Neural Information Processing Systems, Vol. 32* (Vancouver, BC: Curran Associates, Inc.).

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. (2020). "Beyond accuracy: behavioral testing of NLP models with CheckList," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics), 4902–4912. doi: 10.18653/v1/2020.acl-main.442

Schwartz, L., Haley, C., and Tyers, F. (2022). "How to encode arbitrarily complex morphology in word embeddings, no corpus needed," in *Proceedings of the First Workshop on NLP Applications to Field Linguistics* (Gyeongju: International Conference on Computational Linguistics), 64–76.

Srivastava, A., Rastogi, A., Rao, A., Md Shoeb, AA, Abid, A., and Fisch, A. (2023). Beyond the imitation game: quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.* 2835–2856.

Tayyar Madabushi, H., Romain, L., Divjak, D., and Milin, P. (2020). "CxGBERT: BERT meets construction grammar," in *Proceedings of the 28th International Conference on Computational Linguistics* (Barcelona: International Committee on Computational Linguistics), 4020–4032. doi: 10.18653/v1/2020.coling-main.355

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., et al. (2019). "What do you learn from context? Probing for sentence structure in contextualized word representations," in *International Conference on Learning Representations (ICLR), Vol. 7.* (New Orleans, LO).

Tseng, Y.-H., Shih, C.-F., Chen, P.-E., Chou, H.-Y., Ku, M.-C., and Hsieh, S.-K. (2022). "CxLM: a construction and context-aware language model," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (Marseille: European Language Resources Association), 6361–6369.

Vulić, I., Ponti, E. M., Litschko, R., Glavaš, G., and Korhonen, A. (2020). "Probing pretrained language models for lexical semantics," in *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2020*. doi: 10.18653/v1/2020.emnlp-main.586

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., et al. (2020a). BLiMP: the benchmark of linguistic minimal pairs for English. *Trans. Assoc. Comput. Linguist*. 8, 377–392. doi: 10.1162/tacl_a_00321

Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments. *Trans. Assoc. Comput. Linguist*. 7, 625–641. doi: 10.1162/tacl_a_00290

Warstadt, A., Zhang, Y., Li, X., Liu, H., and Bowman, S. R. (2020b). "Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually)," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics), 217–235. doi: 10.18653/v1/2020.emnlp-main.16

Wei, J., Garrette, D., Linzen, T., and Pavlick, E. (2021). "Frequency effects on syntactic rule learning in transformers," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Punta Cana: Association for Computational Linguistics), 932–948. doi: 10.18653/v1/2021.emnlp-main.72

Weissweiler, L., He, T., Otani, N., R. Mortensen, D., Levin, L., and Schütze, H. (2023). "Construction grammar provides unique insight into neural language models," in *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)* (Washington, DC: Association for Computational Linguistics), 85–95.

Weissweiler, L., Hofmann, V., Köksal, A., and Schütze, H. (2022). "The better your syntax, the better your semantics? Probing pretrained language models for the English comparative correlative," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (Abu Dhabi: Association for Computational Linguistics), 10859–10882. doi: 10.18653/v1/2022.emnlp-main.746

Wiedemann, G., Remus, S., Chawla, A., and Biemann, C. (2019). Does Bert make any sense? Interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint: arXiv:1909.10430.*

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint: arXiv:1910.03771*. doi: 10.18653/v1/2020.emnlp-demos.6

Zhan, W. (2017). On theoretical issues in building a knowledge database of Chinese constructions. *J. Chinese Inform. Process*. 31, 230–238.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., et al. (2022). Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068.*

Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. (2021). "Calibrate before use: improving few-shot performance of language models," in *Proceedings of the 38th International Conference on Machine Learning, Vol. 139 of Proceedings of Machine Learning Research*, eds M. Meila and T. Zhang, 12697–12706.

Chapter 7

Corresponds to the following publication:

> Shijia Zhou, **Leonie Weissweiler**, Taiqi He, Hinrich Schütze, David R. Mortensen and Lori Levin (2024). A Computational Methodology for Annotating Rare Linguistic Phenomena with an Application to Probing Caused Motion in LLMs. Accepted at the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING).

Declaration of Co-Authorship Lori Levin suggested the three constructions. Taiqi He created an initial experiment, which he, David Mortensen, Lori Levin and I regularly discussed. Taiqi He wrote up a first draft, which was not accepted for publication. I gave out this research problem to Shijia Zhou, who I closely supervised throughout the process. I suggested the structure of the different experiments and the use of probing classifiers. Shijia Zhou contributed the idea of changing the experimental setup to focus on the addition and removal of words, thereby creating an NLI problem. She annotated data and implemented code. We collaborated closely on conducting the experiment. Shijia Zhou wrote the initial draft of the paper, which I edited. All other authors provided feedback and advice.

Resesarch Context The following chapter is a late result of my time at CMU. While the idea of testing PLMs' ability to distinguish between the semantics of the three constructions was discussed and partially implemented, we never quite found a solid implicit way of evaluating this. That makes this an example of a broader problem in the evaluation of construction meanings: implicit evaluations rely on finding naturally occurring minimal pairs of similar constructions, and then devising a way of testing for the models' understanding or ability to distinguish them, while using an established task and being able to automatically evaluate the output. For this chapter, we found both the naturally occurring pair and a way to test not only its understanding by using an NLI task, but also more fine-grained dimensions of the constructions' meaning. This setup allowed us to test even on closed-source state-of-the-art models like GPT-4.

Constructions Are So Difficult That Even Large Language Models Get Them Right for the Wrong Reasons

Shijia Zhou[1], Leonie Weissweiler[1,3], Taiqi He[2],
Hinrich Schütze[1,3], David R. Mortensen[2], Lori Levin[2]
[1]LMU Munich, [2]LTI, Carnegie Mellon University, [3]Munich Center for Machine Learning
zhou.shijia@campus.lmu.de, weissweiler@cis.lmu.de

Abstract

In this paper, we make a contribution that can be understood from two perspectives: from an NLP perspective, we introduce a small challenge dataset for NLI with large lexical overlap and show that GPT-4 and Llama 2 fail it completely, and then create further challenging subtasks to attempt to explain this failure. From a Computational Linguistics perspective, we identify a group of two constructions with three classes of adjectives which cannot be distinguished by surface features. This enables us to probe for LLM's understanding of these constructions in various ways, and we find that they fail in a variety of ways to distinguish between them, suggesting that they don't adequately represent their meaning.

Keywords: LLMs, construction grammar, semantics, natural language inference, prompting

1. Introduction

This paper focuses on three families of adjective phrases that all contain the intensifier *so* and a finite clausal complement. As shown in Figure 1, they look superficially similar, but have different meanings and syntactic behaviour. The purpose of this paper is to test large language models (LLMs) for their ability to differentiate between them, enabling us to draw larger conclusions about LLM's understanding of *constructions*, meaning-bearing units that contain multiple words, morphemes, and syntactic relationships.

Two of the three types of adjectives lexically license finite complement clauses such as *happy* in *happy that I was freed* and *certain* in *certain that I saw you*, for which the intensifier *so* is optional (*so certain that I saw you* or *certain that I saw you*). When combined with a complement clause, adjectives like *happy* (mostly affective adjectives) differ in meaning from adjectives like *certain* (mostly epistemic adjectives). An affective adjective with a complement usually triggers an inference that the complement caused the feeling expressed by the adjective. For example, in *happy that I was freed*, we can infer that freeing me caused me to be happy. In contrast, in *certain that I saw you*, seeing you is not the cause of my being certain. In fact, I might not have seen you.

The third class is the *Causal Excess Construction* (CEC, Kay and Sag, 2012; Fillmore et al., 2012), where the adjective is interpreted as the cause of the complement. For example in *so happy that I cried*, being happy is interpreted as the cause of my crying. Crucially, the CEC can contain adjectives that do not license a clausal complement on their own. *So big that it fell over* is not acceptable without *so*. **Big that it fell over* is ungrammatical. Conversely, and importantly for the experiments we present, every sentence without *so* cannot be CEC, so although *so happy that I cried* can be CEC, *happy that I cried* cannot be CEC—crying has to be the cause of happiness in the latter example. Kay and Sag classify CEC as a *head-functor* construction because it is the function word *so*, that licenses the clausal complement, not the lexical head. Many adjectives can occur both in the CEC and with lexically licensed complements (*so happy that I cried*). When both complements occur, the lexically licenced one appears closer to the head adjective (*so happy that you are here that I cried* vs **so happy that I cried that you are here.*)

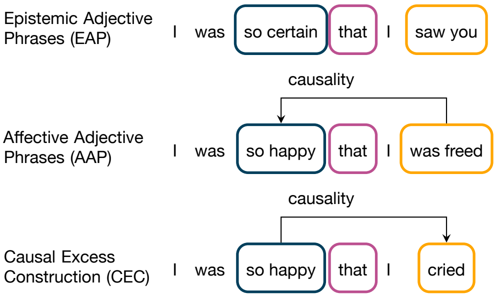

Epistemic Adjective Phrases (EAP)
I
was
so certain
that
I
saw you
causality
Affective Adjective Phrases (AAP)
I
was
so happy
that
I
was freed
causality
Causal Excess Construction (CEC)
I
was
so happy
that
I
cried

Figure 1: Examples of the clausal complement and causal excess constructions. Markers of the constructions are boxed in red.

In this work, we ask: how well can LLMs differentiate between these constructions? They offer an ideal testbed for linguistic probing, since they are structurally identical, which means that there are no exploitable surface cues, and above-baseline performance necessarily stems from a deeper world knowledge, which allows the model to interpret these sentences correctly. Specifically, we investigate GPT-3.5 (OpenAI, 2021), GPT-4 (OpenAI, 2023), OpenAI's `ada-002` and Llama 2 (Touvron et al., 2023) with various questions, for each using both a prompt and a probing classifier.[1] We have observed that, LLMs exhibit limited capability to effectively discriminate between these constructions with a high degree of accuracy, and display a strong bias towards CEC, meaning LLMs tend to judge sentences containing *so... that...* as causal and the main clause being the reason for the subordinate clause. Generally, Llama 2 demonstrates superior performance compared to both GPT-3.5 and GPT-4.

2. Related Work

Our work is situated in the framework of Construction Grammar (CxG), which asserts that there is not a strict division between lexicon and syntax, and that there are meaning-bearing units that consist of multiple words and morphemes, i.e. syntactic structures are paired with meanings just as words are (Croft, 2001; Fillmore et al., 1988; Goldberg, 1995, 2006; Hoffmann and Trousdale, 2013). In this work, we define a construction as a pairing of form and meaning. We are considering AAP, EAP and CEC to be three different constructions. They differ in meaning (affective, epistemic, and excessive states) and in form (a clause that is licensed by a lexical head and a clause that is licensed by the intensifier *so*.)

Recent studies have used CxG to probe the inner workings of LLMs (Weissweiler et al., 2022; Chronis et al., 2023; Mahowald, 2023), as well as made general observations about the compatibility of the theory of CxG with the recent successes of LLMs (Goldberg, To appear; Weissweiler et al., 2023). Most related to our work, McCoy et al. (2019) construct a challenging dataset for Natural Language Inference that uses pairs of sentences with high lexical overlap. They show that the surface similarity of words almost always fools BERT into assuming that one sentence entails the other. Recent work (Si et al., 2022; Basmov et al., 2023) suggests that performance of recent LLMs on the McCoy et al. (2019) data has improved, though it is still far from perfect, which is part of our motivation to create a new challenge dataset.

3. Dataset

Our data collection process takes advantage of Universal Dependencies (Nivre et al., 2020) annotations, which we use for prefiltering before manual annotation. We use SPIKE (Shlain et al., 2020) to access a parsed Wikipedia corpus and a parsed Amazon Reviews corpus. We established that the parse trees of all CEC AAP and EAP constructions have an edge labelled `ccomp` from the adjective to the head verb of the complement clause. We use SPIKE to extract all sentences matching this pattern.

We group the sentences by adjective and manually, where possible, extract a sentence pair where one is CEC and one either AAP or EAP resulting in 111 such pairs. For the adjectives that cannot license a clausal complement, we extract 101 sentences, each with a different adjective. We call this class OCE(only causal excess).

In total, we collected 323 sentences with 212 different adjectives.[2] An example of each sentence type is given in the first row of Table 1.

4. Experiments

We conduct our experiments with two methods. The first approach involves the development of both implicit and explicit prompts.[3] In the second approach, we extract the last-layer embeddings of sentences generated by LLMs and then apply perceptron-based classification to these embeddings, to assess how well the categories are internally represented in the models.

Probing Classifiers We employ a perceptron classifier to test the final layer embeddings of Llama 2, and `ada-002` sentence embeddings. For Llama 2, we use the mean over token embeddings as a sentence embedding, and also test on only adjective embeddings. When the two classes tested are imbalanced, we upsample the smaller class. We group sentences containing the same adjectives together and trained the perceptron using cross-validation over adjectives, meaning that the adjective itself is no longer an exploitable feature. A Bag-of-Words (BoW) model serves as the baseline.

4.1. Natural Language Inference

As shown in Table 2 (Prompts 1-X), we design four prompt variants to test whether LLMs can detect significant changes in meaning when *so* is removed from sentences of the CEC type. For example, *I was so happy that I cried* does not automatically entail *I was happy that I cried*, whereas *I was so happy that I was freed* entails *I was happy that I was freed*.

The results shown in Figure 2 are striking: For CEC and OCE sentences, models achieve accu-

[1] See Appendix Table 7 for hyperparameters

[2] Data and code are available at `https://github.com/shijiazh/Constructions-Are-So-Difficult`

[3] For each prompt, we repeat the mean over six runs of the experiment.

Type	Transformation	CEC	OCE	AAP	EAP
O	Original	I was so happy that I cried.	It was so big that it fell over.	I was so happy that I was freed.	I was so certain that I saw you.
DS	— ‘so’	I was {} happy that I cried.	It was {} big that it fell over .	I was {} happy that I was freed.	I was {} certain that I saw you.
DT	— ‘that’	I was so happy {} I cried.	It was so big {} it fell over .	I was so happy {} I was freed.	I was so certain {} I saw you.
DST	— ‘so’ & ‘that’	I was {} happy {} I cried.	It was {} big {} it fell over .	I was {} happy {} I was freed.	I was {} certain {} I saw you.
AN	+ ‘not’	I was **not** so happy that I cried.	It was **not** so big that it fell over .	I was **not** so happy that I was freed.	I was **not** so certain that I saw you.
P1	main clause	I was so happy.	It was so big.	It was so happy.	I was so certain.
P2	sub. clause	I cried.	It fell over.	I was freed.	I saw you.
Y-N	yes–no question	Did I cry?	Did it fall over?	Was I freed?	Did I see you?

Table 1: Transformations of the dataset with examples

racy below 10%, while demonstrating 90% accuracy for AAP and EAP. It indicates a pronounced bias towards *entailment* in the models, which replicates a behaviour shown for much smaller models: “assume that a premise entails all hypotheses constructed from words in the premise” by McCoy et al. (2019).

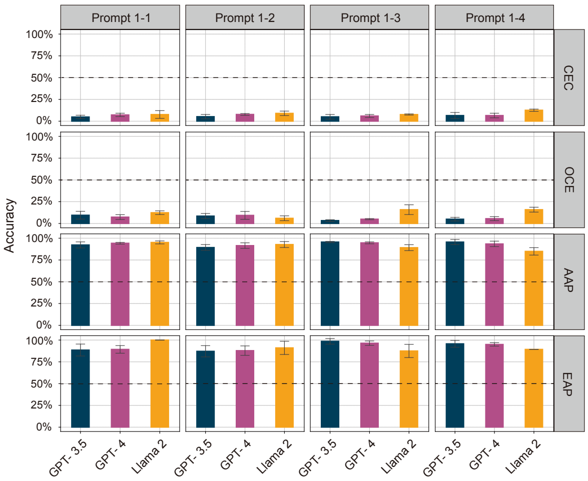


Figure 2: Performance of CEC, OCE, AAP, EAP on the central NLI task. Corresponding gold labels are no entailment, no entailment, entailment, and entailment. All models have a strong bias to answer entailment.

In the following, we investigate three hypothesis that further decompose why LLMs fail at this task: (i) LLMs fail to recognize that removing *so* grammatically disrupts the causal excess construction; (ii) LLMs are unable to identify causality in sentences; (iii) LLMs do not recognize the change in the direction of causality.

4.2. Grammatical Acceptability

Prompting We base our template for grammaticality judgments on Mahowald (2023): 8 pairs of sentence and judgment from the CoLA corpus (Warstadt et al., 2019) are given. The target sentence is inserted, and the model is asked to generate one token, *good* or *bad*.[4]

We test the original sentences (O) and four transformations: DS, DT, DST, and AN. Deleting *that* (DT) or adding *not* (AN) will not affect the grammaticality, whereas removing *so* (DS) from CEC makes the grammaticality debatable and for OCE sentences, renders them ungrammatical.

As can be seen in Table 3, compared to CEC OCE is more likely to be rated as *bad* by both GPT-3.5 and GPT-4, regardless of the transformation. When the gold label for both is *bad*, the gap of their accuracy increases. It demonstrates that GPT models have indeed detected the distinction between OCE and CEC especially regarding their reliance on the grammaticality with *so*. Notably, for GPT models, removing *that* from of AAP and EAP sentences results in more *good*, whereas removing *that* from DS sentences tends to yield more *bad* ratings even though it has no effect on grammaticality.

Llama 2’s answers deviate from that of GPT models. It rates all O, DT, and AN sentences as *good*, which is exactly the gold label, signifying its robust inclination to not only consider sentences featuring *so... that...* as acceptable, but also acknowledge the possibility of omitting *that* in such contexts. However, its performance on DS and DST is far from perfect. Although it achieves higher accuracy for the DS variants of CEC, categorized as *bad*, the 50.03% accuracy is akin to random performance. Additionally, GPT4 performs better on the DS variants of LLC.[5]

4.3. Identifying Causality

Prompting As depicted in Table 4, we devise two prompts to assess the models’ capability to identify causal relationships. The first simply asks about a causal relationship between the main and the subordinate clause, while the second additionally provides the pre-segmented parts. In the EAP category, all models have accuracy below 20%, suggesting a predisposition to infer causality in sentences containing *so... that...* even when the adjective is epistemic. Llama 2 displays a stronger bias, attributing causality to over 90% of sentences in all categories. Combining the previous observation that Llama 2 tends to categorise every sentence containing *so* and *that* as grammatically correct, along with its sensitivity to the

[4] The full prompt is given in the Appendix.

[5] A probing classifier for this prompting task would not be well-defined.

No.		Template
1	1	premise: **O** \n hypothesis: **DS** \n Classify as entailment, no entailment, or contradiction.
	2	premise: **O** \n hypothesis: **DS** \n Classify as entailment, no entailment, or contradiction.
	3	**O** Can we infer that “**DS**”? \n Answer with yes, no or uncertain.
	4	**O** Can we infer that “**DS**”? \n Answer with yes, no or uncertain.
3	1	**O** \n Is there a causal relationship between the main clause and the subordinate clause? \n Answer with yes, no or uncertain.
	2	**O** \n Part1: **P1** \n Part2: **P2** \n Is there a causal relationship between part 1 and part 2? \n Answer with yes, no or uncertain.
4	1	premise: **O** \n hypothesis: **P2** \n Classify as entailment, no entailment, or contradiction.
	2	**O** \n **Y-N** \n Answer with yes, no or uncertain.
	3	**O** \n Part1: **P1** \n Part2: **P2** \n Can we infer that Part1 is the cause of Part2? \n Answer with yes, no or uncertain.
	4	**O** \n Part1: **P1** \n Part2: **P2** \n Can we infer that Part2 is the cause of Part1? \n Answer with yes, no or uncertain.
	5	**O** \n This entails one of two options. \n 1) **P1** because **P2** \n 2) **P2** because **P1** \n Answer with the correct number.

Table 2: Prompt templates of the central NLI task. To test the stability of model responses, we design variants of each prompt, removing only *so* from premise as hypothesis or removing both *so* and *that*.

No.	Model	CEC	OCE	AAP	EAP	Gold Label
O	GPT-3.5	92.43	89.31	80.96	73.57	G|G|G|G
	GPT-4	92.97	89.31	81.93	73.57	
	Llama 2	**100.00**	**100.00**	**100.00**	**100.00**	
DS	GPT-3.5	15.31	36.83	87.23	79.28	B|B|G|G
	GPT-4	15.68	36.43	**88.92**	78.57	
	Llama 2	**23.70**	**39.58**	80.62	**83.34**	
DT	GPT-3.5	89.55	80.40	72.29	60.00	G|G|G|G
	GPT-4	88.83	80.40	70.60	56.43	
	Llama 2	**100.00**	**100.00**	**100.00**	**100.00**	
DST	GPT-3.5	32.61	**57.83**	67.95	65.71	B|B|G|G
	GPT-4	33.33	**57.83**	**68.92**	65.72	
	Llama 2	**50.03**	49.77	60.50	**76.46**	
AN	GPT-3.5	90.63	83.76	74.46	69.29	G|G|G|G
	GPT-4	90.45	84.36	76.14	74.29	
	Llama 2	**100.00**	**100.00**	**100.00**	**100.00**	

Table 3: Accuracy of the grammaticality task. Bold font indicates the models with the highest accuracy for a type and transformation. G: good, B: bad.

No.	Model	CEC	OCE	AAP	EAP	Gold Lab.
3-1	GPT-3.5	60.90	67.33	41.68	**18.57**	Y|Y|Y|N
	GPT-4	58.74	63.37	41.20	15.00	
	Llama 2	**98.65**	**95.05**	**95.18**	08.93	
3-2	GPT-3.5	64.14	54.46	49.15	06.43	Y|Y|Y|N
	GPT-4	65.95	57.03	46.02	04.28	
	Llama 2	**99.10**	**95.54**	**92.78**	**08.93**	

Table 4: Accuracy of the task of identifying causality with different prompts

absence of *so* in CEC, this suggests a limited grasp of semantic nuances and an overreliance on simple lexical cues. GPT models both struggle about equally, with less than 50% accuracy in AAP instances. Even more perplexing, EAP sentences are classified as causal at a significantly higher rate than CEC and OCE.

Probing Classifier The classifiers for CEC vs EAP and AAP vs EAP serve to assess the models' representation of causality. As illustrated in Figure 3, on CEC vs EAP, perceptrons trained on sentence embeddings beat the baseline while those trained on adjective embeddings do not, suggesting that causality is encoded, but not necessarily in the adjective. Interestingly, the adjective perceptron beats the baseline on the AAP vs EAP test, even though the sets of adjectives are mutually exclusive and we perform cross-validation over them, meaning that the only source of information left would be a deeper commonality between them. This suggests that the models may have learned common features for affective and epistemic adjectives, respectively. Furthermore, the result also demonstrates that the distinction between EAP and CEC is more pronounced in the model's perspective compared to the distinction between EAP and AAP, especially for Llama 2 sentence embeddings.

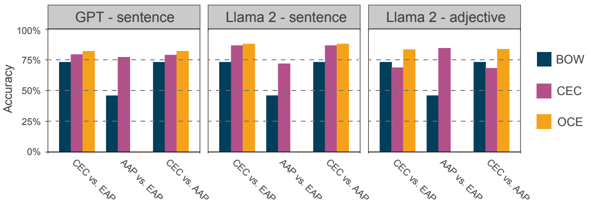


Figure 3: Accuracy of perceptrons trained with different embeddings across three tasks. In all subtasks involving CEC structures, we attempt to replace CEC with OCE. OCE adjectives are mutually exclusive with those in EAP and AAP.

4.4. Direction of Causality

Prompting The negation *not* before *so* influences the truth value of the subclause for CEC but has no influence in AAP. For instance, *He was not so big that he fell.* (CEC) does not imply that he fell, while *He was not so happy that he went.* (LLC) suggests that he went. Asking the model to distinguish between these is equivalent to distinguishing the direction of causality.

Therefore, we have devised an explicit NLI prompt 4-1 (Webson and Pavlick, 2022) and an implicit prompt 4-2 (AN + Y-N) to explore these effects. Accurate answers depend on the models' precise understanding of the causal direction. In addition, we introduced prompt 4-3, where P1 and P2 are provided, and the question is whether P1 is the cause of P2. Prompt 4-4 maintains the same structure but now inquires whether P2 is

Type	Model	CEC	OCE	AAP	Gold Labels
4-1	GPT-3.5	29.19	28.71	62.41	N|N|Y
	GPT-4	29.01	26.93	**62.65**	
	Llama 2	**49.10**	**39.60**	53.62	
4-2	GPT-3.5	2.52	4.55	60.24	N|N|Y
	GPT-4	2.16	4.95	**60.72**	
	Llama 2	**18.47**	**16.34**	40.97	
4-3	GPT-3.5	82.88	74.46	13.25	Y|Y|N
	GPT-4	83.96	77.03	8.91	
	Llama 2	**93.69**	**87.13**	**46.99**	
4-4	GPT-3.5	42.70	50.69	44.09	N|N|Y
	GPT-4	40.18	48.31	47.95	
	Llama 2	**71.17**	**77.23**	**81.92**	
4-5	GPT-3.5	**60.72**	**61.19**	77.59	2)|2)|1)
	GPT-4	54.78	60.80	**79.27**	
	Llama 2	45.49	51.98	78.31	

Table 5: Accuracy of direction of causality task with different prompts. Y: yes/entailment, N: no/contradiction.

the cause of P1. Prompt 4-5 is structured as a multiple-choice format, offering two directions as options.

As can be seen in Table 5, results for prompts 4-1 and 4-2 suggest that they bias all models towards answering *yes* for any sentence. By contrast, prompts 4-3 and 4-4 show a clearer picture, still biased, but Llama 2 scores clearly correlate with the gold label. By comparing these two sets of prompts, we can discern that Llama 2's responses are grounded in an analysis of P1 and P2, rather than simply providing uniform *yes* or *no* answers. Interestingly, prompt 4-5 has elicited better performance from the GPT models in contrast, suggesting that it might be more suited to the multiple-choice answer format. We conclude that all models have some representation of the direction of causality, but it is far from perfect.

Probing Classifier The classifier for CEC vs AAP serve to assess the models' capability to differentiate the direction of causality. The results in Figure 3 are similar to those for identifying causality in section 4.3, with the notable exception of the OCE test set, which is easiest for the adjective classifiers with no obvious explanation.

Figure 3 displays that on CEC vs AAP, similar to CEC vs EAP, models trained with sentence embeddings outperform the baseline, while those trained with adjective embeddings slightly lag behind. Additionally, the perceptron attains the highest accuracy on the OCE test set.

Given that adjectives in OCE can only appear in CEC, while adjectives in CEC can also occur in AAP or EAP, this can be interpreted as OCE's adjective embeddings being more easily identified as belonging to the CEC structure than those of CEC.

5. Conclusion

Overall, our most striking result remains that no LLM performed adequately on our NLI task, and that this result is not sufficiently explained by the mediocre but better-than-baseline performance on the subtasks. Llama 2 performed better in those than the GPT models, but generally, prompting results are often consistently below random and probing classifier results only slightly above baseline. Interestingly, GPT-4 does not perform significantly better than GPT-3.5 at any task.

Both in the central NLI task, and the sub-tasks, all LLMs show bias to offer positive answers. Llama 2 demonstrates a more comprehensive understanding of the grammatical structures within CE, an enhanced ability to identify causality within sentences, and a greater proficiency in ascertaining the direction of causality compared to GPT models. These findings align with our initial observations in the central NLI task.

We have excluded the following from the current work: (1) Extraposition from clausal subject (*That I left was so bad/It was so bad that I left*) (2) CEC meanings with other intensifiers such as *enough* and non-finite clauses (*big enough to fall over*) (3) CEC headed by nouns (*so many people that the police came*). We have also not investigated the CEC in sentences other than copula sentences, or when the CEC adjective is part of a noun phrase (*I met many people so short that they couldn't reach the counter*).

Bibliographical References

Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2023. Chatgpt and simple linguistic inferences: Blind spots and blinds.

Gabriella Chronis, Kyle Mahowald, and Katrin Erk. 2023. A method for studying semantic construal in grammatical constructions with interpretable contextual embedding spaces. *arXiv preprint arXiv:2305.18598*.

William Croft. 2001. *Radical Construction Grammar: Syntactic theory in typological perspective*. Oxford University Press, Oxford, UK.

Charles J. Fillmore, Paul Kay, and Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: the case of 'let alone'. *Language*, 64(3):501–538.

Charles J. Fillmore, Russell R. Lee-Goldman, and Russell Rhodes. 2012. The FrameNet Constructicon. In Hans C. Boas and Ivan A.

Sag, editors, *Sign-Based Construction Grammar*, pages 283–322. CSLI Publications, Stanford, CA.

Adele E. Goldberg. 1995. *Constructions: a construction grammar approach to argument structure*. University of Chicago Press, Chicago.

Adele E. Goldberg. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press, Oxford.

Adele E. Goldberg. To appear. A chat about constructionist approaches and LLMs. *Constructions and Frames*.

Thomas Hoffmann and Graeme Trousdale. 2013. *The Oxford handbook of construction grammar*. Oxford University Press.

Paul Kay and Ivan A Sag. 2012. Cleaning up the big mess: Discontinuous dependencies and complex determiners. In *Sign-based construction grammar*, chapter 5, pages 229–256. Citeseer.

Kyle Mahowald. 2023. A discerning several thousand judgments: GPT-3 rates the article + adjective + numeral + noun construction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 265–273, Dubrovnik, Croatia. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

OpenAI. 2021. GPT-3.5: The Third-and-a-Half-Generation Language Model. `https://api.openai.com/docs/gpt-3.5`.

OpenAI. 2023. GPT-4 Technical Report.

Micah Shlain, Hillel Taub-Tabib, Shoval Sadde, and Yoav Goldberg. 2020. Syntactic search by example. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 17–23, Online. Association for Computational Linguistics.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. In *The Eleventh International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Leonie Weissweiler, Taiqi He, Naoki Otani, David R. Mortensen, Lori Levin, and Hinrich Schütze. 2023. Construction grammar provides unique insight into neural language models. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 85–95, Washington, D.C. Association for Computational Linguistics.

```
Now we are going to say which sentences are
acceptable (i.e., grammatical) and which are not.

Sentence: Flosa has often seen Marn.
Answer: good

Sentence: Chardon sees often Kuru.
Answer: bad

Sentence: Bob walk.
Answer: bad

Sentence: Malevolent floral candy is delicious.
Answer: good

Sentence: The bone chewed the dog.
Answer: good

Sentence: The bone dog the chewed.
Answer: bad

Sentence: I wonder you ate how much.
Answer: bad

Sentence: The fragrant orangutan sings loudest at
Easter.
Answer: good

Sentence: [TEST SENTENCE GOES HERE]
Answer:
Sentence: I wonder you ate how much.
Answer: bad

Sentence: The fragrant orangutan sings loudest at
Easter.
Answer: good

Sentence: [TEST SENTENCE GOES HERE]
Answer:
```

Table 6: Few-shot CoLA prompts template created by Mahowald (2023). We tested 5 types of sentence: O, DS, DT, DST and AN.

	GPT-3.5/GPT-4	Llama 2
temperature	1	0.7
top_p	1	0.95
top_k	-	40
max_tokens	null	512
frequency_penalty	0	-
presence_penalty	0	-

Table 7: Hyperparameters for GPT-3.5, GPT-4, OpenAI's ada-002 and Llama 2

Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your syntax, the better your semantics? probing pretrained language models for the English comparative correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Appendix

Adjective	Type	Sentence
frustrated	CEC	In one XFM show , he became so frustrated that he left the room before Karl finished the segment .
	AAP	I am so frustrated that a $ 500 purchase brought such short lived joy .
proud	CEC	Mandhata had dominated the whole planet and he became so proud that he wanted to rule heaven also .
	AAP	My dad was so proud that his son made ” aliyah ” .
afraid	CEC	One man was so afraid that he camped in the middle of his flock , hoping to evade patrolling cowboys .
	EAP	He was so afraid that rival loyalist inmates wished to kill him inside the prison .
optimistic	CEC	Like Napoleon , Hitler was so optimistic that he falsely believed he ’d make it to Moscow before Winter .
	EAP	I am so optimistic that I made the best choice .
abrupt	OCE	The growth was so abrupt that a village sprang .
beautiful	OCE	The palace was so beautiful that the king of Mengwi heard of Tan Cin Jin .

Table 8: Examples from the collected database. CEC represents causal excess construction, where the adjective is interpreted as the cause of the complement. AAP stands for affective adjective phrases, which usually trigger an inference that the complement caused the feeling expressed by the adjective. EAP stands for epistemic adjective phrases, which lexically liscence non-causal complement.

Chapter 8

Corresponds to the following publication:

> **Leonie Weissweiler**, Nina Böbel, Kirian Guiller, Santiago Herrera, Wesley Scivetti, Arthur Lorenzi, Nurit Melnik, Archna Bhatia, Hinrich Schütze, Lori Levin, Amir Zeldes, Joakim Nivre, William Croft and Nathan Schneider (2024). UCxn: Typologically-Informed Annotation of Constructions atop Universal Dependencies. Accepted at the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING).

Declaration of Co-Authorship The idea of a bottom-up annotation scheme for Constructions in UD was born at the Dagstuhl Seminar "Universals of Linguistic Idiosyncrasy in Multilingual Computational Linguistics", which I attended. A research group was subsequently formed, which I led and organised. Several members joined us after this, which enabled us to build queries for more languages. The queries were written in subgroups, and Nina Böbel and I worked on German together. All authors collaborated on the initial draft of the paper. I co-ordinated the paper writing.

Research Context As argued in Chapter 4, the lack of annotated data is one of the bottlenecks in the evaluation of PLMs for CxG. After having initially considered attempting to organise an entirely new initiative to collect annotated data similar to UD, I was pleasantly surprised, at the Dagstuhl Seminar "Universals of Linguistic Idiosyncrasy in Multilingual Computational Linguistics", to find core members of the UD community interested in adding CxG information. As we will argue, this is mutually beneficial: from the UD perspective, we add information that further delivers on the promise of crosslingual comparability, and in annotating constructions, can even identify annotation errors or gaps in the annotation guidelines. From the CxG perspective, this allows us to benefit from the large existing UD community, with annotated data, established guidelines, and methods of verifying and distributing annotated data. The following chapter presents an initial feasibility study using five constructions annotated in ten languages. The resulting data is already available online and is expected to be added to treebanks in the next UD release.

UCxn: Typologically-Informed Annotation of Constructions atop Universal Dependencies

Leonie Weissweiler,[1] **Nina Böbel,**[2] **Kirian Guiller,**[3] **Santiago Herrera,**[3]
Wesley Scivetti,[4] **Arthur Lorenzi,**[5] **Nurit Melnik,**[6] **Archna Bhatia,**[7]
Hinrich Schütze,[1] **Lori Levin,**[8] **Amir Zeldes,**[4] **Joakim Nivre,**[9] **William Croft,**[10]
Nathan Schneider[4]

[1]LMU Munich & MCML, [2]HHU Düsseldorf, [10]University of New Mexico, [8]Carnegie Mellon University
[3]Paris Nanterre University & CNRS, [4]Georgetown University, [5]Federal University of Juiz de Fora
[6]The Open University of Israel, [7]Institute for Human and Machine Cognition, [9]Uppsala University
weissweiler@cis.lmu.de, nathan.schneider@georgetown.edu

Abstract

The Universal Dependencies (UD) project has created an invaluable collection of treebanks with contributions in over 140 languages. However, the UD annotations do not tell the full story. Grammatical constructions that convey meaning through a particular combination of several morphosyntactic elements—for example, interrogative sentences with special markers and/or word orders—are not labeled holistically. We argue for (i) augmenting UD annotations with an annotation layer for such meaning-bearing grammatical constructions, and (ii) approaching this in a typologically informed way so that morphosyntactic strategies can be compared across languages. As a case study, we consider five construction families in ten languages, identifying instances of each construction in UD treebanks through the use of morphosyntactic patterns. In addition to findings regarding these particular constructions, this study yields important insights on methodology for describing and identifying constructions in language-general and language-particular ways, and lays the foundation for future constructional enrichment of UD treebanks.

Keywords: grammatical constructions, treebanks, Universal Dependencies, typology, corpus annotation

1. Introduction

The notion of a *construction* is an important concept in grammar as it allows for an analysis of patterns of form and function within languages as well as systematic comparisons across languages. Consider the WH-interrogatives in English and Coptic. While English uses a combination of WH-words and word order to encode such questions, Coptic typically leaves WH-words in situ, meaning they occur in the same position as non-interrogative pronouns:[1]

(1)	e-	i-	na-	je	**-pai/-ou**	na-	f	[cop]
	FOC-	I-	FUT-	say	**-it/-what**	to-	him	

‘I shall say **it** to him.’ /
‘**What** shall I say to him?’ (ⲉ-ⲓ-ⲛⲁ-ϫⲉ-ⲟⲩ ⲛⲁ-ϥ)

The notion of a WH-interrogative construction is a shared level of abstraction that underlies the differences between the languages: both languages have conventionalized morphosyntactic means to convey that a piece of information is being sought.

Meaning-bearing grammatical constructions such as interrogatives, conditionals, and resultatives are an object of study within and across languages, and many of these have been the focus of semantic/pragmatic annotation schemes, usually involving manual annotation (§3). Our goal is to annotate them on a large scale across many languages in UD treebanks as automatically and accurately as possible. In this paper, we demonstrate how UD treebanks can be enriched with a layer identifying these larger constructions in a typologically informed way so as to enable cross-linguistic comparisons and typological studies. We present a case study of five construction families and ten languages to illustrate the challenges and opportunities of this approach.

Our goal is challenging because holistic constructions are often not reflected in syntactic labels used in treebanks, which aim to break sentences down into minimal grammatical parts. The UD framework, for example, annotates the individual components of a construction (like the object relation and the interrogative pronoun in (1)) but not the larger whole: there is no ‘interrogative clause’ label in UD. There are other challenges as well. For example, there are many non-canonical and elliptical ways of asking questions in English (e.g., *Can you tell us where?*) and some questions look identical to exclamations. For example, *What stunning views* can be read as a question or exclamation.

Continuing with the example of interrogative constructions highlights some of the challenges, even within English. (2) illustrates a few cases, including

[1] In many cases, prosody or punctuation can also indicate a clause is interrogative. Coptic texts, however, do not use question marks, and e.g. web data contains nonstandard punctuation use (Sanguinetti et al., 2022).

ambiguity with exclamatives, as well as noncanonical kinds of interrogatives involving ellipsis, idioms, and echo questions.

(2) a. WOW what stunning views. [en-EWT]
(Inferred interpretation: 'What stunning views!', not 'What stunning views?')
b. Can you tell us where. [en-EWT]
c. WELL GUESS WHAT!!! [en-EWT]
d. She didn't have what? [en-GUM]

Thus, defining constructions (or families of related constructions) in cross-linguistically comparable ways, determining what is within scope for annotation in a particular language, and reckoning with ambiguity are all significant challenges.

Despite these challenges, we see constructional annotation as a *worthy* mission for the multilingual computational linguistics community, because the empirical work will deepen understanding of constructional phenomena across languages and provide data for further typological studies. It is, in our view, also a *viable* way forward, because the work will draw on the rich ecosystem of UD treebanks and tools in order to add and refine constructional descriptions over time. In addition to offering fuller grammatical descriptions of the treebanked sentences, construction annotations may be used to improve the intra- and interlingual consistency of UD guidelines and data. On the more practical side, construction annotation could be used for downstream tasks like inducing frame-semantic representations, information extraction, or predicting grammatical difficulty for L2 learners depending on strategies found in the L1 language, or for heritage learners depending on strategies found in the dominant language (Bhatia and Montrul, 2020).

To compare across languages, it is necessary to identify patterns larger than a single word or grammatical relation, and to do so in a way that is sensitive to different *morphosyntactic strategies* exhibited by different languages (Haspelmath, 2010; Croft, 2016, 2022). This study is grounded in ideas from Construction Grammar and linguistic typology (§2). Our empirical methodology (§3) is to annotate treebanks in each of 10 languages—English, German, Swedish, French, Spanish, Portuguese, Hindi, Mandarin, Hebrew, and Coptic—for selected constructions by constructing graph pattern queries and matching them against UD trees. The constructions in focus in this paper are interrogatives (§4), existentials (§5), conditionals (§6), resultatives (§7), and noun-adposition-noun combinations where the noun is repeated (NPN; §8). Highlights from our corpus investigations corresponding to each construction are discussed in the respective sections, with a quantitative and qualitative discussion in §9.[2]

2. Background

Universal Dependencies (UD) is a framework for cross-linguistically consistent morphosyntactic annotation, which to date has been applied to over 140 languages (Nivre et al., 2016, 2020; de Marneffe et al., 2021). UD annotation consists of two layers: a morphological layer, where each word is assigned a lemma, a universal part-of-speech tag, and a set of morphological features; and a syntactic layer, where all words are connected into a dependency tree labeled with universal syntactic relations. The syntactic representations are defined to prioritize direct relations between content words, which are most likely to be parallel across different languages; function words are treated as grammatical markers on content words. While the inventories of part-of-speech tags and syntactic relations are fixed to support cross-linguistic comparison, the framework allows language-specific elaboration through the use of language-specific morphological features and subtypes of syntactic relations. CoNLL-U is the standard file format for UD treebanks; trees are encoded in 10 tab-separated columns. The last of these, MISC, is open-ended to support annotations beyond the UD standard itself.

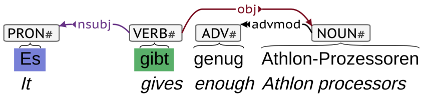
Construction Grammar (CxG) is an approach to linguistic analysis in which the basic unit is a pairing of form and meaning and in which meaning-bearing units can have multiple parts (Construction Elements) (Fillmore et al., 2012; Croft, 2001; Fillmore et al., 1988; Goldberg, 1995, 2006; Hoffmann and Trousdale, 2013). A construction grammar is generally represented as a network indicating taxonomic, partonomic and other relations between constructions, called a Constructicon (Diessel, 2019; Fillmore et al., 2012; Lyngfelt et al., 2018).

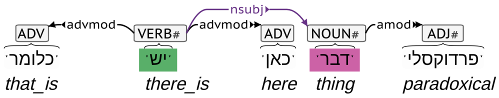
Much work in CxG is done in a single language, where, for example, the English Interrogative Construction is defined by both a particular function and its specific form in English. A broader analysis usually looks at different functions of a constructional form. This is a *semasiological* approach: it starts from a form and examines its possible functions. However, in crosslingual construction analysis, such as linguistic typology, one must find a basis for crosslingual comparison. The basis is generally the function of a construction, because form varies greatly across languages, and many features of morphosyntactic form are defined in language-specific terms such as specific morphemes (English ***do***) or word classes (English Auxiliary).

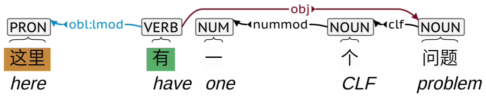
For example, a typology of interrogative constructions compares sentence forms across languages that express the function of a speech act requesting information from an interlocutor. The typological approach is *onomasiological*, starting from a function and considering the various forms realizing it.

[2] Our queries and the final annotated corpora are available at `https://github.com/LeonieWeissweiler/UCxn`.

Language	Instance	Query

German	 PRON#
	nsubj
	VERB#
	obj
	ADV#
	advmod
	NOUN#
	Es
	gibt
	genug
	Athlon-Prozessoren
	It
	gives
	enough
	Athlon processors
		pattern
		EXP [lemma="es"];
		PRED [lemma="geben"];
		PRED -[**nsubj**]-> EXP ;

Hebrew	 ADV
	advmod
	VERB#
	nsubj
	advmod
	ADV
	NOUN#
	amod
	ADJ#
	כלומר
	יש
	כאן
	דבר
	פרדוקסלי
	that_is
	there_is
	here
	thing
	paradoxical
		pattern
		PRED [lemma="יש"];
		PRED -[**nsubj**]-> PIV ;
		without
		LE[lemma="ל"];
		PRED -[obl]->N; N-[case]->LE;

Mandarin	 PRON obl:lmod VERB obj NUM nummod NOUN clf NOUN 这里 有 一 个 问题 here have one CLF problem	**pattern** PRED [form="有"]; PRED -[**obl:lmod**]-> COD ;

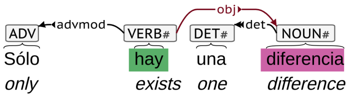
Spanish	 ADV
	advmod
	VERB#
	obj
	DET#
	det
	NOUN#
	Sólo
	hay
	una
	diferencia
	only
	exists
	one
	difference
		pattern
		PRED [lemma="haber"];
		PRED -[**obj**]-> PIV ;
		DET[upos=DET, Definite=Ind];
		PIV -[det]->DET;

Table 1: Existential/presententional construction instances in selected languages and the Grew queries used to identify them. The predicate (PRED), pivot (PIV), coda (COD) and expletive subject (EXP) construction elements and the nsubj, obj and obl:lmod dependency relations are color-coded in the trees and queries.

In typology, a construction as a crosslinguistically valid comparative concept (Haspelmath, 2010) is defined in terms of its function, not its form.[3,4]

Morphemes and word-order can also be described in a crosslinguistically valid fashion. For example, many languages use a special morpheme in interrogative constructions such as Chinese 吗 ***ma***. We can describe this as an "interrogative marker". Other languages change word order in comparison to the declarative construction, albeit in different ways. An interrogative marker or a word order change are two different *strategies* for expressing the interrogative function. We can then describe languages as using the same strategy, or different strategies, for the interrogative construction.

Another important concept is *morphosyntactic recruitment*. If two different constructions such as an existential construction and a possessive construction are morphosyntactically similar, we may say that one construction has *recruited* a strategy from the diachronically or conceptually prior construction, although the directionality or their etymological source may not always be clear.

Related Work Prior work on creating datasets annotated with constructions has been in the form of various Constructicon projects, repositories describing the constructions of a language. One of the first and best known is the Berkeley FrameNet Constructicon for English (Fillmore et al., 2012). Some Constructicons incorporate UD annotations and corpora (for German, Brazilian Portuguese, and Russian; Ziem et al., 2019; Torrent et al., 2018; Bast et al., 2021). While those Constructicons may select individual attestations from corpora to exemplify a construction, in this paper we are concerned with labeling as many instances of the construction as possible in the corpus. Here we take a fundamentally crosslinguistic view of constructions, though the annotation layer could just as well include language-specific constructions. Ultimately we foresee a healthy feedback loop between Constructicon development and corpus enrichment of the kind pursued in this paper.

Construction Grammar has also recently gained popularity in NLP. There have been practical studies using CxG to probe the inner workings of large language models (Weissweiler et al., 2022; Mahowald, 2023), as well as general observations about the compatibility of usage-based constructionist theories with the recent successes of language models (Goldberg, To appear; Weissweiler et al., 2023). Earlier work (Dunn, 2017; Dunietz et al., 2017, 2018; Hwang and Palmer, 2015) in construction-based NLP focused on the annotation, automatic detection and induction of constructions.

3. Methodology

Selection of Constructions For the purpose of cross-lingual comparison, we define constructions in terms of function (e.g., a speech act requesting information), rather than form (e.g., subject-auxiliary inversion). We take a modified onomasiological approach: start from a function, and identify the most conventional forms that express the function. In

[3] In order to distinguish language-specific constructions from constructions as comparative concepts, we follow typological practice and capitalize the names of language-specific constructions.

[4] Some work in CxG such as Hasegawa et al. (2010) is onomasiological and cross-lingual, using frame semantics as the meaning.

many cases, a language conventionally uses more than one strategy for a construction's function. We annotated a few, but not all, of the conventionalized strategies for each construction in each language. Our aim is to see if morphosyntactic queries can detect each strategy in each language, starting only with the information available in UD.

We chose our constructions to be as diverse as possible. We have selected a speech act construction (interrogative), an information structure construction (existential), a complex sentence construction (conditional), an argument structure construction (resultative), and a phrasal construction (NPN). These constructions cover a broad range of specificity, probably annotation complexity, and size. With the NPN construction (Jackendoff, 2008), we examine a strategy, to compare the functions it expresses in the languages in our sample whereas with the other constructions, we examine the functions to compare the strategies recruited in the languages in our sample.

Previous work has explored the relationship between UD annotation and the annotation of (semantically idiosyncratic) multiword expressions (Savary et al., 2023). Here, by contrast, we focus on constructions that are not fully lexically specified—but we share the goal of identifying structures with more to them semantically than meets the eye.

Selection of Languages We select 1 or 2 treebanks for each of a diverse set of languages, with respect to treebank size and typological family. Each language is worked on by at least one linguist who is also a native or proficient level speaker. Our languages and the treebanks we used can be seen in Table 5 in Appendix A. We use UD v2.13.

Although our sample of languages is not representative of global language diversity, covering several languages from several regions ensures that we will cover some variation in strategies.

Identifying Constructions Constructions are defined crosslingually in terms of their *function*, but UD annotates morphosyntactic *form*. For some languages and datasets, we do have functional annotations in addition to syntax trees: for example, the UD English GUM corpus is also annotated with Rhetorical Structure Theory (RST, Mann and Thompson 1988), which identifies pragmatic functions for clauses, including e.g. conditional ones, regardless of how they are expressed. Although we can use this type of information to help identify the scope of ways of expressing a certain meaning or class of meanings in a language, we assume that such annotations are either unavailable for most languages, or do not cover the full breadth of functions whose corresponding constructions we are interested in. Our hypothesis is that, in many cases, we can search for the morphosyntactic *strategies* associated with a construction using UD morphosyntactic annotations and extract tokens of the construction from a treebank with reasonable accuracy.

We test this hypothesis using Grew-match (Guillaume, 2021), which allows us to specify search queries with constraints on sentences and their UD annotations, as shown in Table 1. For each construction, a language may have multiple Grew-match patterns corresponding to multiple morphosyntactic strategies. Grew-match can be combined with Arborator-grew (Guibon et al., 2020) to annotate the trees that it finds.

Annotation atop UD From a technical perspective, we use the optional MISC field (10th column) of the CoNLL-U format used for UD treebanks. The format allows for the introduction of arbitrary key-value annotations. We introduce the key `Cxn`, which is placed on the syntactic head token of the construction from the UD tree perspective, i.e. the highest-ranking node involved in the construction according to the UD tree, or the earliest such node in case of ties. Construction names are given possibly hierarchical names if subtypes are identifiable, such as `Interrogative-Polar-Direct` below, to reflect queries at different levels of granularity.

1	You	you	PRON	...	_
2	have	have	VERB	...	Cxn=Interrogative-Polar-Direct
3	a	a	DET	...	_
4	pencil	pencil	NOUN	...	_
5	?	?	PUNCT	...	_

Appendix B offers a technical specification with full details on the format and naming conventions in our data. The specification also offers the option of annotating *construction elements* in a `CxnElt` field. At present, we annotate only content elements (such as the Protasis and the Apodosis clauses for conditionals; §6), but not functional elements like subordinators that may be strategy-specific.

Next, we proceed construction by construction, first describing a construction in general terms, then highlighting notable findings from querying treebanks.

4. Interrogatives

Typological Overview An interrogative is a speech act construction, expressing a request for information from the addressee. We focus on clauses realizing two major subfunctions: polarity ("Yes/No") questions such as *Is she coming?* and information (content, "WH") questions such as *Who did you see?*. The most common strategies are special prosody, a question marker (see §2) and special verb forms; less common is a change of word order, as in the English examples above. Content questions contain interrogative phrases

		Non-interrog.		Interrog.	
		pre	**post**	**pre**	**post**
English (GUM)	*advmod*	8258	2196	122	1
	nsubj	14512	500	50	0
	obj	265	8889	28	3
	det	15985	36	26	0
	obl	1255	7867	6	1
	ccomp	142	1370	4	0
	xcomp	15	2831	4	0
	other	139	8732	4	1
Coptic	*advmod*	1110	1702	1	3
	nsubj	4844	575	5	2
	obj	2	2585	0	15
	obl	228	4339	35	23
	ccomp	0	750	0	43
	other	2	2478	2	15

Table 2: Pre- and post-posed dependent WH pronouns and non-WH equivalents in EN and CO.

such as *who*, *what* or *which (cat)*; their position varies across languages.

Automatic Annotation Efforts In this section we compare information questions in which the interrogative phrase is placed either in the same position as its non-interrogative counterpart as in (3) or in a different, often fronted position as in (4).

(3) You went where?

(4) Where did you go?

To identify interrogatives, we relied on either the presence of WH items (***what***, ***who***, etc.), word order (in languages using it for marking), as well as the presence of question marks or sentence type annotations where available. In some languages, WH items are identical to indefinite pronouns or free-relative heads (e.g. *I ate what you cooked* is not interrogative, despite containing ***what***), but the UD morphological feature `PronType=Int` helps to disambiguate. We did not see the special verb form strategy in our treebanks.

Table 2 shows pre- and post-posed realization frequencies for different grammatical functions for WH pronouns in interrogatives (i.e. excluding uses such as 'I know who!'), compared to overall usage excluding such pronouns. The table shows the strong preference to front objects in English (28:3 in favor of pre-posed), (other objects appear after their head at a ratio of 265:8889). For other functions, the picture is more complex: interrogative adverbials such as 'when' and 'where' appear almost exclusively preposed, while non-interrogative phrases strongly prefer fronting, but only at a rate of 8258/2196 (79% of cases).

Turning to a different language for comparison, Table 2 shows a rather different picture for Coptic. The tendency for placing subjects before their heads and objects after them is much weaker (5:2, but based on only 7 cases); for adverbial interrogatives, fronting occurs proportionally less than in non-interrogatives. The frequent presence of the Coptic focalizing marker ***ere***, which indicates a contrast with a previously uttered or implied phrase, plays a role in promoting late realization of arguments, above and beyond the tendency for each grammatical function (cf. Green and Reintges 2001).

Takeaways Although typological literature often classifies languages in terms of basic word order or the possibility of word order changes, the actual picture in individual language data is much more complex. We have shown that quantitative analyses with construction-annotated data give a more nuanced picture of how languages realize such word order dependencies in interrogatives.

5. Existentials

Typological Overview Existentials assert the existence (or not) of an entity ('pivot'), almost always indefinite, and usually specified in a location ('coda'), as in *There are yaks in Tibet*. This function is closely related to the presentational function, introducing a referent, as in *There's a yak on the road*. In our language sample, the two functions are expressed using the same strategies, such that their distinction may be largely context-dependent. For this reason, we consider here both existentials and presentatives.

Languages vary with respect to the predicate that they use in the existential. One class of languages employ a construction-specific lexeme, such as Swedish ***finnas***. In Coptic there are lexicalized negative and positive existence predicates, ⲟⲩⲛ ***oun*** and ⲙⲙⲛ ***mmn***. Historically, predicative possession used the same items, but through lexicalization, the possessive versions are now lexically distinct from the existentials.

The relationship between existence and possession also has synchronic manifestations. Our sample includes languages that use a possession verb as the predicate in an existential, one predicate to express both existence and possession, such as ***ter*** 'to have' in Brazilian Portuguese, French ***avoir*** in the phrase *il y a*, or the Mandarin predicate 有 ***yǒu***. This duality is also found in Hebrew, where possession is expressed by adding a possessive dative argument to the existential construction (5).

(5) hayu (la-nu) kama taxanot ba-derex
were.3P (to-us) few stops.PF in.the-way
'There were/We had a few stops on the way.'
(היו (לנו) כמה תחנות בדרך) [he-HTB]

An additional existential strategy shares a copula predicate with the predicational locative construction. In Hebrew, the copula היה ***haya*** is used in past and future tense existentials (*hayu* in (5) is the inflected form). For Mandarin, the use of the copula 是 ***shì*** is an alternative to the lexicalized existential predicate 有 ***yǒu***. The link between the locative construction and the existentials is also found in locatives that grammaticalized into unique existential forms such as English ***There('s)*** or French ***y*** (in *il y a*).

Finally, the existential predicates ***haber*** and ***haver*** in Spanish and European Portuguese, respectively, also function as auxiliaries and modals, similarly to the English ***have***, modulo possession.

The argument structure of existential predicates is not uniform crosslinguistically, with pivots exhibiting different degrees of subjecthood properties (Keenan, 1976). This diversity is manifested in the UD annotation. In one class of languages, no argument is identified as `nsubj` and the pivot is attached as `obj` in UD. This is the case in Spanish and Mandarin (see Table 1).

Other languages identify the pivot as `nsubj`. This is the case in Hebrew, where the copular predicate standardly exhibits agreement with the pivot, as in (5). However, unlike typical subjects, the Hebrew pivot appears post-verbally, does not always trigger agreement, and in informal speech may receive accusative marking, if definite. Likewise, in Coptic the pivot is `nsubj` in postverbal position, though the adverb ***there*** is added in around 5% of cases in the UD data with no clear antecedent.

A different strategy involves employing an expletive as a co-argument to the pivot. This is found in our language sample in French and English (6) and in German (Table 1).

(6) il y a une salle à l'étage
it there has a room upstairs
'There is a room upstairs.' [fr-GSD]

Here, too, UD annotations vary across languages. In the English treebank the pivot is attached as `nsubj` and ***there*** as `expl`. In French, the expletive ***y*** is `expl:comp` and the pivot is `obj`. In German the expletive ***es*** is `nsubj` and the pivot is `obj`.

Automatic Annotation Efforts Our languages vary in the difficulty in identifying existential constructions. The easiest cases were those in which a construction-specific lexical item is employed(e.g., the lexicalized existential predicates in Coptic and Swedish). In French, instances of the existentials are identified by queries which target the construction-specific cooccurrence of the clitic ***y*** and the verb ***avoir*** in a `comp:expl` relation.

The more challenging cases are those in which the elements which encode existence are multifunctional. In some treebanks, this challenge is overcome by construction-specific annotations. Thus, for example, in the Hebrew HTB the predicate היה ***haya*** is annotated as `HebExistential=Yes` where it is used in its existential function.

When disambiguating annotations are not available, the queries rely on other distributional properties of the construction to avoid false positives. In French, the queries only target indefinite pivots, excluding definite determiners and numerals. In Hebrew, the queries exclude instances where the predicate has a `obl` dependent with a dative case marker, i.e., the possessor (see query in Table 1). Furthermore, to distinguish between the predicational and existential functions of היה ***haya*** in UD_Hebrew-IAHLTwiki (Zeldes et al., 2022), where this information is not annotated, the query targets only post-verbal `nsubj` dependents.

Takeaways Lexical items that are associated with the existential construction are often shared with other constructions. For this reason, in order to maximize accuracy the queries cannot only rely on these lexical items but also target morphosyntactic properties and dependency relations.

6. Conditionals

Typological Overview A conditional construction is a complex sentence construction describing a broadly "causal" link between the two states of affairs, the protasis (condition) and the apodosis (consequence) (Comrie, 1986, pp. 81–82). The strategies for conditional constructions are largely the typical ones for complex sentences in general (Croft, 2022, pp. 532–34). The construction may be an adverbial subordinate construction or a coordinate construction. The clauses may be balanced (identical in form to a declarative main clause) or deranked (one clause, usually the protasis, is in a distinct form, with a special verb form and other differences). There may be a subordinating conjunction such as ***if***, or rarely a change in word order, as in English *Had he stayed, he would have seen it.* The nonfactual nature of conditionals may manifest in irrealis or subjunctive verb forms.

Automatic Annotation Efforts Common strategies for conditionals are the use of a subordinating conjunction as in German *Wenn die Möglichkeit da ist* (lit. if the opportunity there is) (3291 instances in German-HDT, 240 in Swedish-Talbanken, 243 in Hindi-HDT, 495 in English-GUM), or word order inversion as in Swedish *Har du god kondition* (lit. have you good condition; if you are in good shape) (1182 instances in German-HDT, 68 in Swedish-Talbanken, 7 in English-GUM). A very different strategy involves conditional circumfixes. In Coptic ***e--šan*** is a circumfix that conveys conditionality (CD) and applies to the pronominal subject of the conditional clause so that *e-f-šan-eibe* (CD-he-CD-thirst) means *If he is thirsty.*

Our investigation of conditionals has shown that it may not be possible, using the information available in UD, to create queries that accurately retrieve conditional sentences. There are three sources of difficulty: (1) the need for information that is not yet encoded in UD, (2) subordinating conjunctions and clause types that are not exclusively used in conditional constructions, and (3) the variety of subordinating conjunctions and other strategies that are used to express the conditional construction.

Conditional subordinating conjunctions can be divided into: simple subordinating conjunctions like ***agar***/***yadi*** (Hindi) ('if'); complex subordinating conjunctions like ***förutsatt att*** (Swedish) ('provided that'); and V2 sentence embedders like ***angenommen*** (German) ('presumed') (Breindl et al., 2014). Complex subordinating conjunctions and V2 sentence embedders are problematic in German HDT because the part of speech is not *conjunction* and the dependency label is not *mark* (or *fixed expression* as in Swedish), making it necessary to search for the lemma of the connector, which results in many false positives. These conjunctions are also not labeled as *mark*.

In Germanic languages, conditional constructions without subordinating conjunctions usually express the protasis as verb-initial clauses that precede the main clause. In Swedish and German, any verb can be used in a verb-initial protasis clause. In English, however, it is restricted to certain auxiliaries (e.g. ***Had*** *I gone, I would have seen you*).

While the subtypes of English conditionals (e.g. neutral or negative epistemic stance) require many search queries but are in principle findable with UD, this is not the case in German. A major problem for German conditionals—especially with regard to semantic and syntactic subcategorization—was posed by the inadequate mood annotations. German HDT does not annotate conditional or potential verb forms and marks most verb forms as indicative, even when there is a clear conditional or subjunctive structure. It is therefore not possible to search for semantic subcategories based on different mood annotations in HDT, although verb mood is the most common indication of grouping conditionals in German (Schierholz and Uzonyi, 2022).

Takeaways The conditional strategies are in principle searchable, although writing these rules requires an exhaustive study of the phenomenon in each language. Search requirements vary in complexity depending on the depth of the underlying linguistic analyses of the phenomenon. Annotation practices in the treebank complicate the search process and even make some distinctions impossible.

7. Resultatives

Typological Overview From a functional perspective the resultative construction expresses an event with two subevents: a *dynamic* subevent such as ***paint*** and a *resulting state* subevent such as ***red*** in *They painted the door red*.

(7) They painted the door **red**.

The English resultative construction is a prime example of an argument structure construction (Hovav and Levin, 2001; Goldberg and Jackendoff, 2004). A basic transitive clause describing an event is augmented with a secondary predicate describing the result state of a participant, but there are many strategies to express this function. English, for example, also uses adverbial subordination of the dynamic event: *The door was red as a result of their painting it* or *I flattened the metal by hammering*. In our study of resultatives, we are only considering cases where the language provides a conventionalized form-meaning pairing for expressing the complex event composed of a manner of action and a result state as a complex predicate, resulting in some interesting challenges.

Automatic Annotation Efforts In the sample languages, we encountered several challenges. First, in some languages a resultative conceptualization is lacking: they do not combine a dynamic event with a stative result event into a complex predicate. In Hebrew, the most natural way of expressing the painting event literally translates as 'They painted the door in red' (the result expressed with an oblique marked by the prepositional prefix ***be-*** 'in'). In Hindi, complex predicates expressing a cause-result relation have a dynamic event as a result (8). We consider these languages as lacking the resultative construction as defined above.

(8) ... ki veh duSman ko **maar**
... that it enemy ACC **hit**
bhagaa-ye [hi-HUTB]
run.CAUS-SUBJ
'... that it beat and chase away the enemy.'

Second, in several languages the overwhelming number of "resultative" constructions were of the form ['make/do' X STATE], where the dynamic event is the causative verb 'make/do', e.g., Hindi ***kar*** 'do', Swedish ***göra*** 'make/do', or German ***machen*** as in *Nvidia machts möglich. (Nvidia makes it possible)*. This construction is analyzed as the causative of a stative event, and is excluded from the resultative category. In the German and Swedish treebanks, removal of the causative left few or no examples of genuine resultatives.

Third, in some languages, the UD annotation of the resultative construction is indistinguishable from another construction such as depictive secondary predication. For example, *I hammered the metal flat* (resultative) has the same structure as *I left the door open* (depictive). It was necessary in English for queries to incorporate lexical lists of predicates licensing the construction, in order to disambiguate

Lang.	SU	CO	OP	PR	QU
COP	+	–	+	–	(+)
EN	+	+	+	+	+
FR	+	(+)	+	+	(+)
DE	+	–	+	+	+
HE	+	+	+	+	(+)
HI	(?)	(?)	(?)	–	–
ZH	(?)	–	–	–	–
PT	+	+	+	+	(+)
ES	+	+	+	+	(+)
SV	+	(+)	(+)	+	+

Table 3: Semantic categories of NPN and their cross-linguistic attestation in UD treebanks. – means that the target meaning is not possible in the language. (+) signals that the meaning is possible but not attested in the UD treebanks. (?) means that the existence of this meaning is unclear, see footnote 6. Succession: SU, Comparison: CO, Opposition: OP, Proximity: PR, Quantification: QU

from other sentences with similar UD structures, at the expense of generalizability to predicates that have not been seen in the resultative construction.

Finally, Chinese has a very productive resultative construction (9), which is already annotated in the treebank Chinese-HK (Wong et al., 2017) with a label specifically designed for resultative complements: `compound:vv`. They are trivially extracted by querying for that dependency relation.

(9) wǒ **qiāo píng** le dīngzi [zh-HK]
1SG **hit** **flat** PERF nail
'I hammered the nail flat.' (我敲平了钉子.)

Takeaways In summary, the attempt to annotate the resultative construction has shown us several difficulties: annotating a construction where boundaries are in dispute within the literature, which might not even exist in all languages depending on the definition, and where considerable linguistic expertise and manual effort is required to write a comprehensive set of rules, indicating the need for collaboration among theoretical linguists, corpus linguists, typologists and computational linguists. Efforts such as ours can reveal constructions that need further linguistic investigation, or can help solidify linguistic consensus on the definition of the construction.

8. NPN

Typological Overview With the preceding four constructions, we took an onomasiological approach, examining them cross-linguistically on a functional basis. Most work in CxG, however, takes a semasiological (form-first) approach to characterizing a formal pattern and its function(s), usually within a single language. In our terms, this approach starts with a strategy and examines the range of functions using that strategy. The UD framework offers a common vocabulary for describing formal categories of morphology, parts of speech, and grammatical relations across languages. In this section, we consider how a semasiological (or strategy-based) inquiry can be conducted cross-linguistically using UD corpora. As a case study, we look at the "NPN" strategy, in which a meaning related to quantification or iteration is expressed with a repeated noun and an adposition or case marker on the second noun. Examples in English include *day after day, shoulder to shoulder, box upon box,* etc. While infrequent and often a source of idioms, this strategy recurs across many languages (Postma, 1995; Matsuyama, 2004; Jackendoff, 2008; König and Moyse-Faurie, 2009; Roch et al., 2010; Pskit, 2015, 2017; Kinn, 2022).[5]

Automatic Annotation Efforts We find examples of NPN strategies across 8/10 languages.[6] In our queries, we limit ourselves to instances where the two Ns are the same lemma, though there are related NPN uses where the two Ns are not the same (Jackendoff, 2008). A few examples of NPN from our treebanks are presented in (10).

(10) PT: ***frente a frente*** 'face to face' (lit. 'front to front'), FR: ***jour pour jour*** 'to the day' (lit. 'day for day'), SV: ***steg för steg*** 'step by step' (lit. 'step for step'), HE: ***mila be-mila*** 'word for word' (lit. 'word in word')

In terms of morphosyntactic form, NPN strategies are well captured by our queries because of the strict precedence relationship between the constituent elements. We find that there is considerable variability in whether NPNs are analyzed as fixed expressions in UD (using the `fixed` relation type), or whether the second N is analyzed as an `nmod` of the first N.[7]

The semantics of NPN have been well investigated in previous literature (Jackendoff, 2008; Roch et al., 2010; Sommerer and Baumann, 2021; Kinn, 2022). We find that most of the previously proposed semantic subcategories emerge in our languages. Following the categorization and discussion in Jackendoff (2008) and later works, we find the following semantic subtypes of NPN: SUCCESSION (*hour af-*

[5] The studies cover Dutch, English, French, German, Norwegian, Japanese, Mandarin, Polish, and Spanish.

[6] We did not find any attestations of NPN in the Chinese or Hindi treebanks. It is unclear whether NPN is productive in these languages, but we are aware of expressions that might qualify: e.g. Mandarin ***yī tiān bǐ yī tiān*** and Hindi ***din ba din*** (both 'day by day').

[7] We restrict our queries to exclude cases where the first N is marked by another adposition, because we find that in many languages the PNPN strategy (*from time to time*) has a different range of meanings than the NPN strategy. We also exclude cases where nouns are modified with adjectives, as these are extremely rare.

Lang.	Interrogative (§4)	Existential (§5)	Conditional (§6)	Resultative (§7)	NPN (§8)	total sent.	total tokens
EN	599; 544	319; 478 (f)	256; 516 (D)	64; 64 (H, D)	11; 19	11k; 17k	187k; 254k
DE	8376 (H)	3480 (H)	4437 (A,H)	D	40	190k	3.5m
SV	234	10	306 (H)	D	7	6k	96k
FR	669	109 (F)	296 (F)	D	12	16k	400k
ES	887	160 (F)	515 (F)	D	37	18k	567k
PT	293 (A)	358 (F)	116	D	7	9k	227k
HI	186	2058 (F)	288 (A)	D	0	16k	351k
ZH	148	58 (F)	31	78 (D)	–	1k	9k
HE	225; 35	335; 230	192; 57	D	9; 11	6k; 5k	160k; 140k
CO	158	78	186	D	2	2k	55k

Table 4: Counts of identified construction instances by treebank, along with qualifications: definitional issues (D), UD annotation errors (A), occasional false positives (f), frequent false positives (F), unattested strategies (H). A dash means the construction does not exist in that language. The two numbers for EN and HE represent the two treebanks for each (cf. Table 5).

ter hour), COMPARISON (*man for man*), OPPOSITION (*brother against brother*), PROXIMITY (*hand in hand*) and QUANTIFICATION (particularly of a large quantity, *snacks upon snacks*). Qualitatively, we noticed that the SUCCESSION submeaning was most prevalent across our treebanks, and OPPOSITION is typically restricted to body parts, as in (10). Table 3 summarizes our empirical findings by semantic subtype and language.

Takeaways Using a strategy, like NPN, as the basis of typological comparison is not without issue (Croft, 2022); however, we do find considerable functional overlap in terms of the meanings which are conveyed by the NPN strategy in our language sample. Notably, NPN is the only investigated construction/strategy for which the query is almost universal across languages, meaning that it is the most well-integrated with UD: if the promise is universality across languages, then ideally a query would also work across all languages. It makes perfect sense that this only works with strategies, which are defined by their form, and not for constructions, which are defined by their meaning, as UD itself focuses on form.

9. Survey Summary

Our 5 case studies have surveyed constructions and strategies in 10 languages. Table 4 provides a quantitative summary in terms of matched instances per treebank. Treebanks ranged in size from 9k to 3.5m tokens; in some cases, the scale was too small for a robust set of results. NPN was particularly sparse—this is simply a rare strategy.

Table 4 also provides a qualitative summary of some of the major kinds of issues encountered: definitional issues (D), annotation errors in the treebank (A), unavoidable occasional false positives (f), many false positives due to overlap with another construction (F), and unattested strategies for which at least one query returned 0 examples (H). For most of the languages, we abandoned attempts to quantify resultatives given the definitional challenges. Note that some of the larger treebanks had unattested strategies (H)—this is not necessarily because of a problem with the treebanks, but reflects that more effort was put into writing queries for long-tail strategies in those languages.

We are pleased to see that UD annotation errors (A) were not a major source of difficulty for most of the treebanks examined. On the other hand, many constructions were fundamentally difficult to circumscribe (D) or distinguish from other constructions given the available UD annotations (F). These may necessitate human annotation and/or supplementary information from semantic analyzers.

For English and Hebrew, where we consulted two treebanks, we can see some differences in the construction counts that are not explained by the size of the treebank but rather by the domain. This underscores the importance of domain diversity in empirical studies of constructions.

10. Conclusion and Future Work

In this work, we have provided a case study of annotating constructions in UD treebanks. We developed automatic annotation queries for ten languages and five construction families, and detailed our results and takeaways. Overall, we find that annotating constructions is feasible with a mix of automatic and manual efforts, and that with typologically-based construction definitions, the annotations support crosslinguistic quantitative studies. The next step is to scale up our annotation methodology to more languages and constructions, possibly with the aid of construction parsers (and/or UD parsers to produce larger-scale silver treebanks for investigating rare constructions). Beyond the created resources, these efforts may prompt improvements to the UD annotation guidelines and to language-specific Constructicons. Crucially, this work has been a first attempt at bringing two important frameworks together. We aim to gather feedback and input from the community to further our goal of integrating constructions fully with UD.

Acknowledgments

This work was initiated by the Dagstuhl Seminar 23191 "Universals of Linguistic Idiosyncrasy in Multilingual Computational Linguistics" https://www.dagstuhl.de/en/seminars/seminar-calendar/seminar-details/23191. We are grateful for Grew infrastructure support from Bruno Guillaume, and for feedback from members of the NERT lab at Georgetown and anonymous reviewers. This work was supported in part by Israeli Ministry of Science and Technology grant No. 0002336 (Nurit Melnik, PI) and NSF award IIS-2144881 (Nathan Schneider, PI).

Bibliographical References

Radovan Bast, Anna Endresen, Laura A. Janda, Marianne Lund, Olga Lyashevskaya, James McDonald, Daria Mordashova, Tore Nesset, Ekaterina Rakhilina, Francis M. Tyers, and Valentina Zhukova. 2021. The Russian Constructicon. An electronic database of the Russian grammatical constructions. Available at https://constructicon.github.io/russian/.

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. 2017. The Hindi/Urdu treebank project. In *Handbook of Linguistic Annotation*. Springer Press.

Archna Bhatia and Silvina Montrul. 2020. Comprehension of differential object marking by Hindi heritage speakers. In A. Mardale and S. Montrul, editors, *The Acquisition of Differential Object Marking*, pages 261–281. John Benjamins Publishing Company.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies treebank for German. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France. Association for Computational Linguistics.

Eva Breindl, Anna Volodina, and Ulrich Hermann Waßner. 2014. *Handbuch der deutschen Konnektoren 2: Semantik der deutschen Satzverknüpfer*, volume Band 13 of *Schriften des Instituts für Deutsche Sprache*. De Gruyter, Berlin and München and Boston.

Bernard Comrie. 1986. Conditionals: a typology. In E. C. Traugott, A. ter Meulen, J. S. Reilly, and C. A. Ferguson, editors, *On Conditionals*, pages 77–99. Cambridge University Press.

William Croft. 2001. *Radical Construction Grammar: Syntactic theory in typological perspective*. Oxford University Press, Oxford, UK.

William Croft. 2016. Comparative concepts and language-specific categories: Theory and practice. *Linguistic Typology*, 20(2):377–393. Publisher: De Gruyter Mouton.

William Croft. 2022. *Morphosyntax: Constructions of the world's languages*. Cambridge University Press.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Holger Diessel. 2019. *The Grammar Network: How Linguistic Structure is Shaped by Language Use*. Cambridge University Press, Cambridge.

Jesse Dunietz, Jaime Carbonell, and Lori Levin. 2018. DeepCx: A transition-based approach for shallow semantic parsing with complex constructional triggers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1691–1701, Brussels, Belgium. Association for Computational Linguistics.

Jesse Dunietz, Lori Levin, and Jaime Carbonell. 2017. The BECauSE corpus 2.0: Annotating causality and overlapping relations. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 95–104, Valencia, Spain. Association for Computational Linguistics.

Jonathan Dunn. 2017. Computational learning of construction grammars. *Language and cognition*, 9(2):254–292.

Jan Einarsson. 1976. *Talbankens skriftspråkskonkordans*. Institutionen för nordiska språk, Lunds universitet.

Charles J. Fillmore, Paul Kay, and Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: the case of 'let alone'. *Language*, 64(3):501–538.

Charles J. Fillmore, Russell R. Lee-Goldman, and Russell Rhodes. 2012. The FrameNet Constructicon. In Hans C. Boas and Ivan A. Sag, editors, *Sign-Based Construction Grammar*, pages 283–322. CSLI Publications, Stanford, CA.

Adele E. Goldberg. 1995. *Constructions: a construction grammar approach to argument structure*. University of Chicago Press, Chicago.

Adele E. Goldberg. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press, Oxford.

Adele E. Goldberg. To appear. A chat about constructionist approaches and LLMs. *Constructions and Frames*.

Adele E Goldberg and Ray Jackendoff. 2004. The english resultative as a family of constructions. *language*, pages 532–568.

Melanie Green and Chris H. Reintges. 2001. Syntactic anchoring in Hausa and Coptic wh-constructions. *Proceedings of the Annual Meeting of the Berkeley Linguistics Society*, 27(2).

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. 2020. When collaborative treebank curation meets graph grammars. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5293–5302, Marseille, France. European Language Resources Association.

Bruno Guillaume. 2021. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics.

Bruno Guillaume, Marie-Catherine de Marneffe, and Guy Perrier. 2019. Conversion et améliorations de corpus du français annotés en Universal Dependencies [conversion and improvement of Universal Dependencies French corpora]. *Traitement Automatique des Langues*, 60(2):71–95.

Yoko Hasegawa, Russell Lee-Goldman, Kyoko Hirose Ohara, Seiko Fujii, and Charles J. Fillmore. 2010. On expressing measurement and comparison in English and Japanese. In Hans C. Boas, editor, *Contrastive Studies in Construction Grammar*, pages 169–200. John Benjamins, Amsterdam.

Martin Haspelmath. 2010. Comparative concepts and descriptive categories in crosslinguistic studies. *Language*, 86(3):663–687.

Thomas Hoffmann and Graeme Trousdale. 2013. *The Oxford handbook of construction grammar*. Oxford University Press.

Malka Rappaport Hovav and Beth Levin. 2001. An event structure account of english resultatives. *Language*, pages 766–797.

Jena D. Hwang and Martha Palmer. 2015. Identification of caused motion construction. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 51–60, Denver, Colorado. Association for Computational Linguistics.

Ray Jackendoff. 2008. "Construction after Construction" and Its Theoretical Challenges. *Language*, 84(1):8–28.

Edward Keenan. 1976. Towards a universal definition of subject. In Charles N. Li, editor, *Subject and Topic*, pages 303–334. Academic Press New York, New York.

Torodd Kinn. 2022. Regular and compositional aspects of NPN constructions. *Journal of Linguistics*, 58(1):1–35.

Ekkehard König and Claire Moyse-Faurie. 2009. *Spatial reciprocity: between grammar and lexis*, page 57–68. De Gruyter Mouton.

Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago Timponi Torrent. 2018. *Constructicography: Constructicon development across languages*, volume 22. John Benjamins Publishing Company.

Kyle Mahowald. 2023. A discerning several thousand judgments: GPT-3 rates the article + adjective + numeral + noun construction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 265–273, Dubrovnik, Croatia. Association for Computational Linguistics.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Tetsuya Matsuyama. 2004. The N After N Construction. *English Linguistics*, 21(1):55–84.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pages 1659–1666.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Dan Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, pages 4034–4043.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Gertjan Postma. 1995. Zero Semantics — The syntactic encoding of quantificational meaning. *Linguistics in the Netherlands*, 12:175–190.

Wiktor Pskit. 2015. *The Categorial Status and Internal Structure of NPN Forms in English*, page 27–42. Cambridge Scholars Publishing.

Wiktor Pskit. 2017. *Linguistic and philosophical approaches to NPN structures*, page 93–110. Wydawnictwo Uniwersytetu.

Alexandre Rademaker, Fabricio Chalub, Livy Real, Cláudia Freitas, Eckhard Bick, and Valeria de Paiva. 2017. Universal Dependencies for Portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 197–206, Pisa,Italy. Linköping University Electronic Press.

Claudia Roch, Katja Keßelmeier, and Antje Muller. 2010. Productivity of NPN sequences in German, English, French, and Spanish. In *Proceedings of the Conference on Natural Language Processing 2010*, page 158–163, Saarbrücken, Germany.

Shoval Sade, Amit Seker, and Reut Tsarfaty. 2018. The Hebrew Universal Dependency treebank: Past present and future. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Manuela Sanguinetti, Lauren Cassidy, Cristina Bosco, Özlem Çetinoğlu, Alessandra Teresa Cignarella, Teresa Lynn, Ines Rehbein, Josef Ruppenhofer, Djamé Seddah, and Amir Zeldes. 2022. Treebanking user-generated content: a ud based overview of guidelines, corpora and unified recommendations. *Language Resources and Evaluation*, 57:493–544.

Agata Savary, Sara Stymne, Verginica Barbu Mititelu, Nathan Schneider, Carlos Ramisch, and Joakim Nivre. 2023. PARSEME Meets Universal Dependencies: Getting on the same page in representing multiword expressions. *Northern European Journal of Language Technology*, 9(1).

Stefan J. Schierholz and Pál Uzonyi, editors. 2022. *Grammatik: Band 2: Syntax*, volume Bd. 1.2 of *Wörterbücher zur Sprach- und Kommunikationswissenschaft*. De Gruyter, Berlin and Boston.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Lotte Sommerer and Andreas Baumann. 2021. Of absent mothers, strong sisters and peculiar daughters: The constructional network of english NPN constructions. *Cognitive Linguistics*, 32(1):97–131.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Tiago Timponi Torrent, Ely Edison Matos, Ludmila Meireles Lage, Adrieli Laviola, Tatiane da Silva Tavares, Vânia Gomes de Almeida, and Natália Sathler Sigiliano. 2018. Towards continuity between the lexicon and the constructicon in FrameNet Brasil. In Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago Timponi Torrent, editors, *Constructicography: Constructicon development across languages*, pages 107–140. John Benjamins, Amsterdam.

Leonie Weissweiler, Taiqi He, Naoki Otani, David R. Mortensen, Lori Levin, and Hinrich Schütze. 2023. Construction grammar provides unique insight into neural language models. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 85–95, Washington, D.C. Association for Computational Linguistics.

Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your syntax, the better your semantics? probing pretrained language models for the English comparative correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tak-sum Wong, Kim Gerdes, Herman Leung, and John Lee. 2017. Quantitative comparative syntax on the Cantonese-Mandarin parallel dependency treebank. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 266–275, Pisa, Italy. Linköping University Electronic Press.

Amir Zeldes. 2017. The GUM corpus: creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Amir Zeldes and Mitchell Abrams. 2018. The Coptic Universal Dependency Treebank. In *Proc. of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 192–201, Brussels, Belgium.

Amir Zeldes, Nick Howell, Noam Ordan, and Yifat Ben Moshe. 2022. A second wave of UD Hebrew treebanking and cross-domain parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4331–4344, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexander Ziem, Johanna Flick, and Phillip Sandkühler. 2019. The german constructicon project: Framework, methodology, resources. *Lexicographica*, 35(2019):61–86.

A. List of Treebanks

An overview of the treebanks used in this work along with their total number of sentences is provided in Table 5. The genres covered by each treebank are shown in Table 6.

B. Technical Specification

The specification of the data format and naming conventions for constructions and construction elements appears below.

Lang	Treebanks	Num. Sents
EN	EWT, GUM (Silveira et al., 2014; Zeldes, 2017)	16,662; 10,761
DE	HDT (Borges Völker et al., 2019)	189,928
SV	Talbanken (Einarsson, 1976; Nivre et al., 2006)	6,026
FR	GSD (Guillaume et al., 2019)	16,342
ES	AnCora (Taulé et al., 2008)	17,662
PT	Bosque (Rademaker et al., 2017)	9,357
HI	HUTB (Bhat et al., 2017)	15,649
ZH	Chinese-HK (Wong et al., 2017)	1,004
HE	HTB, IAHLTwiki (Sade et al., 2018; Zeldes et al., 2022)	6,143; 5,000
CO	Coptic Scriptorium (Zeldes and Abrams, 2018)	2k

Table 5: UD treebanks used in our crosslinguistic study. Some cover specific varieties, e.g., AnCora represents European Spanish, whereas Bosque covers both European and Brazilian Portuguese. Chinese is limited to Mandarin. Coptic (Sahidic) is the only historical language.

	EN	DE	SV	FR	ES	PT	HI	ZH	HE	CO
academic	+									
bible										+
blog	+			+						
e-mail	+									
fiction	+									+
government	+									
grammar examples										
learner essays										
legal										
medical										
news	+	+	+	+	+	+	+		+	
nonfiction	+	+	+							+
poetry										
reviews	+			+						
social	+									
spoken	+							+		
web	+	+								
wiki	+			+					+	

Table 6: Genres covered by the UD treebanks used in the paper.

Chapter 9

Corresponds to the following publication:

> **Leonie Weissweiler**, Abdullatif Köksal, Hinrich Schütze (2025). Hybrid Human-LLM Corpus Construction and LLM Evaluation for the Caused-Motion Construction. Northern European Journal of Language Technology.

Declaration of Co-Authorship: I conceived the research contribution. I implemented the data collection pipeline and manually annotated the necessary data. Abdullatif Köksal advised me on the choice of LLMs to evaluate and ran my evaluation code using these models. I wrote the initial draft of the paper, and all authors provided feedback and helped to refine the draft.

Research Context After developing the initial phase of a community project for CxG annotation, I was still in need of data for my next evaluation project, which was not on one of the five constructions included. While the UCxn project will hopefully develop into a large database, I believe that immediate solutions for finding data are still needed. With the aim of evaluating LLMs' understanding of the caused-motion construction, after initial experiments showed them to be struggling, I expanded on the methodology for annotation from Chapter 5 and included queries on dependency trees, very similar to those used in Chapter 8. However, the caused-motion construction is so semantically complex that it would be classified as having a high number of false positives, by the standards laid out in the last chapter. The key idea is therefore to further reduce false positives by prompt-engineering ChatGPT, and then to ensure the quality of the final data, finally perform manual annotation. Using the collected data, we will now go on to test LLMs on the subtleties of the construction.

Hybrid Human-LLM Corpus Construction and LLM Evaluation for the Caused-Motion Construction

Leonie Weissweiler, Abdullatif Köksal, Hinrich Schütze
LMU Munich & Munich Center for Machine Learning
weissweiler@cis.lmu.de

Abstract The caused-motion construction (CMC, "She sneezed the foam off her cappuccino") is one of the most well-studied constructions in Construction Grammar (CxG). It is a prime example for describing how constructions must carry meaning, as otherwise the fact that "sneeze" in this context takes two arguments and causes motion cannot be explained. We form the hypothesis that this remains challenging even for state-of-the-art Large Language Models (LLMs), for which we devise a test based on substituting the verb with a prototypical motion verb. To be able to perform this test at a statistically significant scale, in the absence of adequate CxG corpora, we develop a novel pipeline of NLP-assisted collection of linguistically annotated text. We show how dependency parsing and LLMs can be used to significantly reduce annotation cost and thus enable the annotation of rare phenomena at scale. We then evaluate OpenAI, Gemma3, Llama3, OLMo2, Mistral and Aya models for their understanding of the CMC using the newly collected corpus. We find that most models struggle with understanding the motion component that the CMC adds to a sentence.

1 Introduction

(1) She sneezed the foam off her cappuccino.

(2) They laughed him off the stage.

These are two examples of the caused-motion construction (CMC) in which the verb behaves unusually: *sneeze* and *laugh* typically do not take multiple arguments, nor do they typically convey that something was moved by sneezing/laughing. This poses a challenge to any naive form of lexical semantics: it would not make sense for someone writing a dictionary to include, for each intransitive verb, the meaning and valency of the CMC. Almost any verb can appear in the CMC as long as we can imagine a scenario in which the action it describes causes motion. The fact that humans easily understand the CMC showcases a main feature of Construction Grammar (Croft, 2001; Goldberg, 1995): the meaning is attached to the construction itself, and not the verb. Putting the verb into this construction adds the new meaning and valency. This is one reason that constructions pose a challenge to Large Language Models (LLMs), as they would have to learn to attach the meaning to this construction and retrieve it when necessary. Its extreme rarity and productivity makes it impossible to memorise all instances and memorisation would not be sufficient because the meaning shift to the verb is creative and is influenced by the specific context.

The research questions of this paper therefore are: Have LLMs learned the meaning of the CMC and how can we construct the resources needed to determine the status of CMC in LLMs?

We first address the second question, of collecting data for this at scale. This is challenging for several reasons. First, the CMC is a very rare phenomenon. Second, we are mostly interested in instances that are non-prototypical, i.e., where the verb does not typically encode motion, unlike e.g. 'kick' or 'throw'. Third, this construction cannot be automatically identified using only syntactic criteria: words might be in the correct syntactic slots required by the CMC, but not create a CMC reading if the semantics of the sentence do not fit. For example, "I would take that into account" is structurally identical to the examples above, but nothing is moving.

This shows that there is a crucial semantic component. The rarity makes it very costly to manually sift through a corpus to collect a dataset of the CMC, while the semantic complexity makes it infeasible to do so fully automatically.

In this way, we consider the CMC exemplary of rare phenomena of language that have been largely set aside in Computational Linguistics and in recent evaluation of LLMs in particular. This may be due to them being considered the *periphery* of language, rather than the core (Chomsky, 1993), or simply due to the described

https://doi.org/10.3384/nejlt.2000-1533.2025.5256

difficulty in finding appropriate data to investigate both the phenomena and their representation in LLMs. However, it is our point of view that as the performance of such models increases across the board, it is vital to turn to "edge cases" to accurately identify performance gaps. This is particularly important as rare phenomena may be indicators of systematic underlying problems of an NLP paradigm.

To study rare phenomena, we need natural data for them at scale. To this end, in section 3 we propose a novel annotation pipeline that combines dependency parsing with the use of LLMs. The aim of our pipeline is to minimise the cost of running the LLM and compensating human annotators, while maximising the number of positive, manually verified, linguistically diverse instances in the dataset.

After creating our corpus, we now return to our aim of evaluating state-of-the-art LLMs for their understanding of the CMC, as an example of a semantically challenging "edge case".

In Section 4, we therefore develop a test for different LLMs' understanding of the CMC, by giving an instance and asking if the direct object is physically moving. We then replace the verb (e.g., "sneeze") by a prototypical one that always encodes motion (e.g., "throw") and ask the model again if the direct object is moving. We expect models that do not fully understand the CMC to fail to consistently answer both questions with "yes". We observe that models struggle with this task to varying degrees.

We make three main contributions:

- We propose a hybrid human-LLM corpus construction method and show its effectiveness for the CMC, an extremely rare phenomenon. We discuss how our design and our guidelines can be applied to data collection needs for other linguistic phenomena.
- We release a corpus of manually verified instances of the CMC of 500 sentences.[1]
- We evaluate different sizes of Llama3, Gemma3, OLMo2, Mistral, Aya, and OpenAI models on their understanding of the CMC and find that most models struggle.

2 Related Work

Evaluation of LLMs' Understanding of Constructions. Tayyar Madabushi et al. (2020) conclude that BERT (Devlin et al., 2019) can classify whether two sentences contain instances of the same construction. Tseng et al. (2022) show that LMs have higher prediction accuracy on fixed than on variable syntactic slots and infer that LMs acquire constructional knowledge (i.e., they understand the "syntactic context" needed to identify a fixed slot). Weissweiler et al. (2022) find that LLMs reliably discriminate instances of the English Comparative Correlative (CC) from superficially similar contexts. However, LLMs do not produce correct inferences from them, i.e., they do not understand its meaning.

Zhou et al. (2024) evaluate LMs' understanding of the causal excess construction by contrasting it with two constructions of similar structure, and using the LMs' ability to distinguish between them as a proxy for measuring their understanding. They find that even large models like GPT-4 perform poorly on this. By contrast, Rozner et al. (2025a), using the same dataset among others, investigate smaller masked language models. They do not test understanding but rather probe the internal representations of the output layer to recover systematic differences between the constructions, showing that distinguishing between them is possible. Rozner et al. (2025b) repeat this experiment with BabyLM models and find that even they are capable of picking up many constructions, providing valuable evidence about construction learning with developmentally plausible amounts of data.

Bonial and Tayyar Madabushi (2024) compile a corpus of examples from several constructions, including the 52 caused-motion sentences collected from the Abstract Meaning Representation (AMR) dataset (Banarescu et al., 2013). They evaluate GPT-4 and GPT-3.5 on their ability to pick out three caused-motion sentences from among a larger set, and find that performance does not exceed 60%. However, it should be noted that this was metalinguistic prompting, relying on a model's understanding of the term 'caused-motion', which many humans may also be unfamiliar with.

Most related to this work, Li et al. (2022) probe for LMs' handling of four Argument Structure Constructions (ASCs): ditransitive, resultative, caused-motion, and removal. They adapt the findings of Bencini and Goldberg (2000), who used a sentence sorting task to determine whether human participants perceive the argument structure or the verb as the main factor in the sentence meaning. They find that, while human participants prefer sorting by the construction more if they are more proficient English speakers, language models show the same effect in relation to training data size. In a second experiment, they then insert random verbs that are incompatible with one of the constructions, and measure the Euclidean distance between the verbs' contextual embedding and that of a verb that is prototypical for the construction. They demonstrate that

[1] Code and data are provided on https://github.com/LeonieWeissweiler/CausedMotion

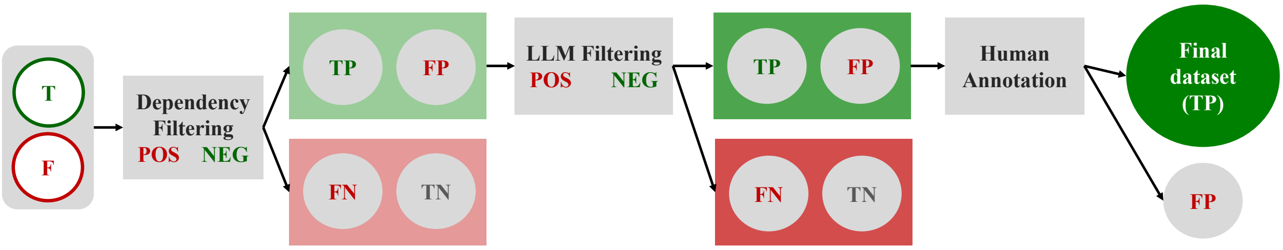


Figure 1: Flowchart of our annotation pipeline. For details of each step refer to §3.

construction information is picked up by the model, as the contextual embedding of the verb is brought closer to the corresponding prototypical verb embedding.

Mahowald (2023) investigates GPT-3's (Brown et al., 2020) understanding of the English Adjective-Article-Numeral-Noun construction (AANN), assessing its grasp of the construction's semantic and syntactic constraints. Utilising a few-shot prompt based on the CoLA corpus of linguistic acceptability (Warstadt et al., 2019), he creates artificial AANN variants as probing data. GPT-3's performance on the linguistic acceptability task is found to align with human judgments across most conditions. More recently, Misra and Mahowald (2024) investigate the same construction for smaller models trained on the BabyLM corpus (Warstadt et al., 2023) and show how its learning is supported by more frequent, smaller constructions. In a similar vein, Scivetti et al. (2025) investigate how well BabyLM size models acquire the let-alone construction.

Linguistic Annotation with LLMs Since the release of ChatGPT, numerous papers have proposed to use it or similar LLMs as an annotator. Gilardi et al. (2023) find that ChatGPT outperforms crowd-workers on tasks such as topic detection. Yu et al. (2023) and Savelka and Ashley (2023) evaluate the accuracy of GPT-3.5 and GPT-4 against human annotators, while Koptyra et al. (2023) annotate a corpus of data labelled for emotion by ChatGPT, but acknowledge its lower accuracy compared to a human-annotated version. In the area of Construction Grammar, Torrent et al. (2023) use ChatGPT to generate novel instances of constructions.

Most related to our work are papers that propose a cooperation between the LLM and the human annotator. Holter and Ell (2023) create a small gold standard for industry requirements by generating an initial parse tree with GPT-3 and then correcting it with a human annotator. Pangakis et al. (2023) investigate LLM annotation performance on 27 different tasks in two steps. First, annotators compile a codebook of annotation guidelines, which is then given to the LLM as help for annotation, and then the codebook is refined by the annotators in a second step. However, they find little to no improvement from the second step. Gray et al. (2023) make an LLM pre-generate labels for legal text analytics tasks which are then corrected by human annotators, but find that this does not speed up the annotation process.

In contrast, our work proposes a hybrid human-LLM pipeline that minimizes the cost of dataset creation. We emphasise prompt design and engineering, a critical factor in effective use of LLMs.

Computational Approaches to Argument Structure Constructions. In addition to the probing work discussed above, ASCs have also been studied from a computational perspective. Kyle and Sung (2023) leverage a UD-parsed corpus as well as FrameNet (Fillmore et al., 2012) semantic labelling to annotate a range of ASCs.

Hwang and Palmer (2015) identify CMCs and four different subtypes based on linguistic features. Some of these are automatically generated, but others are gold annotations. This limits the applicability to large, unannotated corpora.

Hwang and Kim (2023) conduct an automatic analysis of constructional diversity to predict ESL speakers' language proficiency. Similar to our first filtering step, they perform an automatic dependency parse and then identify a range of constructions, including the CMC, using a decision tree built on the parse. They do not employ any further filtering.

3 Data Collection

Concept of the CMC In collecting a dataset of CMC instances, we must first find a working definition of the CMC to guide our automatic and manual annotation. While we base our definition on that of Goldberg (1992), we also restrict it further to include only sentences in which the object is physically moving. This is not meant as a universal definition of the CMC, but rather as one that suits the needs of our project, as we later ask LLMs if the direct object is moving and where. We therefore make no definitive statement as to whether metaphorical movement (*I laughed myself off the chair*), the electronic movement of data (*I sent him an email*), or movement involving a metaphysical location (*She sneezed herself out of existence*) constitute instances of the CMC.

Data Collection Pipeline Our aim is to investigate how well the caused-motion construction is learned by LLMs, for which we require a dataset of caused-motion sentences, which should be natural and therefore sourced from text. The simplest version of this would be to have human annotators sift through a corpus and extract all caused-motion sentences. This would be very expensive, as we assume caused-motion sentences to be quite rare. On the other hand, they are so semantically complex that we cannot simply use automated filtering, e.g. based on dependencies. We therefore propose a hybrid approach combining linguistic resources, an LLM, and an expert annotator.

Our key idea is that data collection will proceed in a pipeline, where a corpus is first filtered using dependency parsing and the syntactic constraints of the CMC, the output set of sentences is further filtered with prompt-based classification using an LLM, and the sentences which it labels as positive are then manually annotated by a human. Each step in the pipeline is meant to further concentrate the rate of instances in the corpus that will then be manually annotated, therefore reducing total annotation effort.

The main cost of data collection is the cost of the LLM API and for human annotators. We assume that any expenses for linguistic resources and the computational infrastructure (not relevant to running LLMs) at our disposal are negligible in comparison. *Our aim is to minimise the cost for the LLM and annotators while maximising the number of positive, manually verified, diverse instances.*

We propose a way of computing the cost for this problem setting and a pipeline for producing a novel linguistic resource while minimising cost.

Our main goal is to minimise the cost per confirmed CMC sentence; however, we also have a secondary goal: the final set of sentences should be diverse. Regardless of the specific goals of the linguistic researcher, it is unlikely that they would be served by a set of sentences that do not represent the true diversity of the CMC. Extreme cost-minimising measures – such as making the dependency filtering rules described in §3.1 too strict or asking the LLM to provide examples of the CMC – would therefore be counterproductive.

The baseline here is to take an annotator, give them a corpus, set them on the task of reading through it and marking all sentences that contain instances of the CMC. As the corpus contains very few true positives, this would be highly costly. We therefore turn to dependency parsing with spaCy (Honnibal et al., 2020) for prefiltering. We select the reddit corpus (Baumgartner et al., 2020), with the motivation that it will contain a high rate of creative language usage, aiding our goal of finding as many non-prototypical CMC instances as possible.

class	PR	RE	F1	n
True	79.76	97.10	87.58	69
False	75.00	26.09	38.71	23
Avg	77.38	61.59	63.15	92

Table 1: Accuracies of the dependency filtering based on the total set of positive and negative instances from Goldberg (1992). We focus on maximising Recall (RE) of the True class, to minimise the number of potential CMC sentences that are lost before human annotation, achieving 97%.

3.1 Step 1: Dependency Parsing

Figure 1 shows our pipeline. In the first step, we dependency-parse the corpus and apply a pattern to filter out all sentences that, with high likelihood, are not instances of the phenomenon.

For this dependency annotation, we could rely on annotated treebanks such as Universal Dependencies (de Marneffe et al., 2021). But to find a diverse and sufficiently large set of instances, particularly in languages other than English, available treebanks may not be large enough for the rare phenomenon that we are targeting.

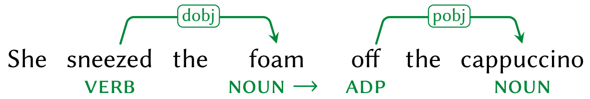
We therefore turn to automated dependency parsing to annotate large amounts of data, which we can run by using minimal computational resources without the need for GPUs.

After dependency parsing, we want filters that preserve the diversity of the found sentences. We therefore design subtree filters that preserve recall above all else. This is especially advisable as parsing will lead to some parsing errors that we want to be tolerant of, and as CMC sentences are rare, they are more likely to be parsed incorrectly.

To design the pattern, we start with a list of gold instances taken from Goldberg (1992), which we parse with the spaCy toolkit.[2] The instances are positive and negative examples for the CMC. On the basis of their dependency parses, we develop dependency constraints as a filter for our dependency-parsed sentences. Specifically, we iterate over the verbs in a sentence, then look for a direct object or a recursive dependent of the direct object, e.g. an adjective, immediately following the verb. In the position immediately following, we check for an adposition, while taking into account that it may comprise several tokens. We do not impose constraints on the dependency between adposition and

[2] version 3.2.0

prepositional object, as we have found these to be especially vulnerable to parsing errors. We then look for a pobj-dependent of this adposition.



We design the subtree to optimise recall with reasonable precision, following the overall goal of losing as few sentences as possible in the pipeline to maximise final dataset diversity.

We then evaluate its recall and precision on this small development set, comprising the total sum of positive and negative CMC instances given in Goldberg (1992), and report on the results in Table 1. Our filter achieves 97.10 % recall for true CMC instances, minimising the number of sentences lost in this step.

This filtering step also allows us to extract the location of the potential CMC instance and its parts as a side product of the filtering step: We extract the sentence, the lemmatised verb, direct object, preposition, and prepositional object, as well as their positions in the sentence.

3.2 Step 2: Selection of Sentences for Classification

Given that we now have a lot of dependency-filtered data and limited resources for classification, we want to select the optimal set of sentences for this classification, in order to optimise several criteria for our final dataset. As the dataset will form a challenging evaluation set for LLMs, the most important of these criteria is that the dataset contains as many verbs as possible that do not usually contain motion. Even though we consider sentences like “I throw the ball” instances of the CMC, they would not challenge a model’s understanding, as “throw” already encodes motion. As a proxy for this, we sort verbs by how frequently they are used intransitively, with the idea that these would make for less prototypical CMC sentences.

We compute statistics about the verbs with UD. Specifically, we merge the English treebanks EWT (Silveira et al., 2014), GUM (Zeldes, 2017), GUM reddit (Behzad and Zeldes, 2020), LinES (Ahrenberg, 2007), partTUT (Sanguinetti and Bosco, 2015), PUD (Zeman et al., 2017), and GENTLE (Aoyama et al., 2023), and then for each verb, we compute the ratio of how often that verb has an object. We then go through the dependency-filtered dataset from the last step and sort by this ratio. This has the added benefit of removing verbs that never appeared in UD as lemmata, which removes noise from the reddit dataset.

3.3 Step 3: Prompt-based Few-shot Classification with an LLM

Goals Even after dependency-based filtering, the positive instances would still be very rare in the output, and it is therefore not feasible that the output is directly annotated by a human. We therefore introduce a further filtering step with an LLM to “concentrate” the positive instances even more, i.e. we want the LLM to remove most negative instances while keeping as many positive instances as possible. The remaining data can then be cost-effectively annotated by the human annotator. The aim is to reduce the cost per instance (i.e., cost per true positive, TP) as much as possible.

There are two components of the cost: the cost of querying the LLM and the cost of human annotation. Our two key ideas are:

- We consider the two costs jointly and optimise the pipeline for overall lowest cost per TP.
- Design and selection of the prompting setup (including the prompt, the choice of model, how many times it’s run, etc.) used with the API is a major determinant for the cost of the pipeline. We propose a workflow for creating effective prompting setups.

A particular prompting setup may require many tokens in total, thereby incurring a higher API cost. But it may also have high accuracy, thereby reducing the cost of human annotation. We jointly consider both cost components when designing and selecting prompting setups.

Development Set For creating the development set V, we manually annotate 500 (183 positive, 317 negative) sentences from the output of the dependency filtering step. To ensure that V is both diverse and relevant, we group the prefiltered dataset by verb, and starting with the highest-frequency verbs, take at most 5 positive and 5 negative sentences from every verb, where no preposition appears twice in either the positive or the negative sentences selected. We choose 25 shots from each class to be included as examples in the prompt, which are not used for V.

Minimising the cost per true positive Given this development set, let $J(C_{\mathrm{HR}}, C_{\mathrm{API}}, i)$ be the cost per true positive where C_{HR} is the human annotation cost per sentence, C_{API} is the cost of processing an input/output token with the API and i (for instruction) is a prompting setup. We can then estimate $J(C_{\mathrm{HR}}, C_{\mathrm{API}}, i)$, the cost per true positive, as follows:

$$\frac{C_{\mathrm{API}}t(V,i) + C_{\mathrm{HR}}(\mathrm{TP}(V,i) + \mathrm{FP}(V,i))}{\mathrm{TP}(V,i)} \qquad (1)$$

				Sent's to Annotate			**Total Cost**		
P	**Details**	**Prec.**	**Rec.**	**LLM**	**Human**	**API**	C_{HR}**=$.002**	C_{HR}**=$.006**	C_{HR}**=$.5**
1	Base (4o-mini)	0.486	0.582	3535	1719	0.01	3.46	10.3	860
2	1 + repeat sentence with json	0.459	0.656	3320	1524	0.04	3.13	9.2	762
3	2 + reason	0.470	0.662	3217	1512	0.07	3.15	9.2	756
4	3 + structured information	0.648	0.621	2483	1610	0.07	3.33	9.8	805
5	4 + sentence	0.519	0.664	2900	1505	0.07	3.13	9.2	753
6	4 + cmc string	0.393	0.462	5507	2167	0.09	4.44	13.1	1083
7	4 + cmc string continuous	0.536	0.681	2744	1469	0.06	3.06	8.9	735
8	6 + sentence	0.536	0.658	2839	1520	0.06	3.15	9.2	760
9	7 + sentence	0.579	0.658	2622	1519	0.06	3.14	9.2	760
10	4 + few shots	0.557	0.600	2990	1667	0.08	3.46	10.1	833
11	10 + explanations	0.694	0.608	2371	1646	0.06	3.39	10.0	823
12	11 + all shots	0.710	0.653	2155	1531	0.07	3.18	9.3	766
13	**12 + only 10 samples**	0.721	0.714	1943	1402	0.11	**3.02**	8.6	701
14	12 + only 1 sample	0.552	0.789	2296	1267	0.76	4.17	9.2	635
15	12 + only 5 samples	0.639	0.713	2192	1402	0.19	3.18	8.8	701
16	12 + only 25 samples	0.738	0.678	1998	1474	0.08	3.09	9.0	737
17	14 + new few-shots	0.552	0.789	2296	1267	0.76	4.17	9.2	635
18	17 + alternating shots	0.588	0.805	2114	1243	0.70	4.04	9.0	623
19	17 + grouped shots	0.448	0.796	2803	1256	0.93	4.53	9.6	630
20	19 + majority vote	0.486	0.840	2449	1191	2.44	7.97	12.7	601
21	19 on o3-mini	0.913	0.856	1280	1168	4.91	13.83	18.5	595
22	21 + 100 samples	0.760	0.874	1506	1144	0.67	3.90	8.5	574
23	21 + 250 samples	0.820	0.806	1513	1240	0.52	3.64	8.6	621
24	21 + 50 samples	0.803	0.865	1440	1156	0.80	4.20	8.8	580
25	21 + 25 samples	0.798	0.864	1451	1158	0.83	4.27	8.9	581
26	24 + majority vote	0.803	0.891	1397	1122	2.42	8.13	12.6	567
27	24 on 4o	0.814	0.837	1467	1195	0.75	4.10	8.9	599
28	24 - sentence	0.787	0.878	1447	1139	0.75	4.07	8.6	571
29	27 - sentence	0.803	0.821	1516	1218	0.54	3.65	8.5	610
30	**28 - reason**	0.803	0.891	1397	1122	0.70	3.96	**8.4**	563
31	29 - reason	0.760	0.790	1667	1266	0.60	3.82	8.9	634
32	**22 on o1**	0.880	0.920	1235	1087	5.79	16.72	21.1	**558**
33	32 + 50 samples	0.891	0.916	1226	1092	7.10	19.94	24.3	564
34	33 + majority vote	0.869	0.952	1209	1050	22.30	60.10	64.3	583
-	Human only	-	-	-	2732	0.00	5.46	16.4	1366

Table 2: A comparison of all prompting setups for different values of C_{HR}. **P** = Prompting Setup. We give numbers (sentences that need to be annotated by LLM/human) for a scenario in which the desired size of the final resource (output of pipeline when applied to the raw corpus) is $N = 1000$. The human baseline depends solely on the rate of TPs (which is higher here than for the raw corpus to be processed by the pipeline as the development set contains more positive instances). The different values of C_{HR} were chosen to highlight the different scenarios in which the three best prompting setups, 13, 30, and 32, are each optimal.

where we process the development set using the API and prompting setup i and record: $\text{TP}(V, i)$, the number of true positives, $\text{FP}(V, i))$, the number of false positives, and $t(V, i)$, the sum of the number of tokens input to the API and the number of tokens returned by the API.

We create a variety of different prompting setups (where with prompting setup we refer to a combination of prompt, model, and other configurations like majority voting) i and then select our final prompting setup i' as the one with the lowest per-TP cost:

$$i' = \text{argmin}_i J(C_{\text{HR}}, C_{\text{API}}, i)$$

Determining the size of the input corpus To compile our CMC dataset, we set a target number of $\text{TP}_{\text{req}} = 292$ instances of the CMC, to bring the total up to 500 by later adding the manually annotated positive development instances and the positive few-shots. After selecting a prompting setup i and determining $\text{TP}(V, i)$ on the development set, we can estimate the size N of the input corpus that will result in a set of TP_{req} instances to be output by the pipeline as:

$$N := |V| \frac{\text{TP}_{\text{req}}}{\text{TP}(V, i)}$$

Iterative Prompting Setup Development We start with a simple base prompting setup and iteratively attempt improvements to it. The total cost of this experimentation was about $22. The full details of all attempted prompting setups are given in the appendix in Section A. We test four models from OpenAI of those available in February 2025: 4o-mini, 4o, o3-mini, and o1. For this experiment, we use sampling with temperature=1.0 and top_p=1.0.[3]

During prompt development, we do not have a good estimation of the human annotator cost, as we will ultimately annotate the sentences ourselves. We, however, assume that C_{HR} should be at least $0.001, which means that we can determine many prompting setup improvements to be clear improvements and only have to consider the cost tradeoff for some.

We start with a simple prompting setup that gives no few-shot examples and asks for sentence IDs and classifications in a csv codeblock, classifying 50 sentences at a time with 4o-mini. The instruction remains the same throughout and can be seen in the prompt example in Table 3. We achieve straightforward improvements by making the model repeat the sentence (and therefore giving the output as a json object to avoid confusion over commas), but not with having 4o-mini give a reason for its decision. We then try out different combinations of giving the entire sentence, only the substring containing the core CMC, and the structured information given by the dependency parsing step. We add few shots and hand-written explanations for our labels for them. We also vary the number of samples, increase the number of few-shots, and reorder them. We then add majority voting after running each sentence 3 times, and try out different numbers of sentences to be classified for each prompt. During this process, we also switch to the more expensive models o3-mini, 4o, and o1. The final optimal prompting setup depends on the human annotation cost. In Figure 2, we visualise with grey vertical lines where one prompting setup "overtakes" another, meaning the human annotation cost per sentence where the optimal prompting setup changes. We then show example total cost figures for three reasonable values in between these change points in Table 2, revealing that the best prompting setups are 13, 30, and 32, depending on human annotation cost.

As our **final prompting setup**, we select prompting setup 30 as it is a good tradeoff between API cost and human cost.

3.4 Final Dataset Collection

In combination with the 183 positive instances from the development set, and an additional 25 positive instances from the few shots, we now set out to annotate additional data using our pipeline, to reach a final dataset of 500 hand-annotated CMC instances. To this end, we classify an additional 9,046 sentences with prompting setup 30, with approximately 3.6 USD in API costs. 598 of these (6.6%) are classified as positive by the model. We annotate these by hand, resulting in 292 positive and 396 negative instances, which gives the prompting setup a precision of 48.83% in practice. We see the reason for this lower precision mostly in the fact that the concentration of true positives was likely much lower in the data processed here, than in the development set, which was chosen to have many diverse CMC instances. Examples for sentences in the final dataset are given in Table 4.

4 Evaluation of LLMs' Understanding of the CMC

4.1 Methods

The goal of our evaluation is to assess different LLMs for their understanding of the CMC. The performance reached by the prompts in the data collection phase is not a suitable measure for this, since it relied on metalinguistic prompting and few-shots.

Our LLM evaluation setup in this section differs from prompting setup evaluation as we do not explicitly refer to the "caused-motion construction", but rather prompt implicitly for the model's understanding of the situation described. The key idea is that in a CMC sentence, something is always physically moving, even if the verb (e.g., "sneeze") does not indicate this. The distinction between prototypical vs. non-prototypical instances is crucial here: for prototypical CMC instances ("throw", "kick"), the verb already conveys the meaning component of motion while for non-prototypical CMC instances ("sneeze", "laugh") it does not and the LLM has to infer the additional meaning component of motion from the construction.

Our setup is to ask "In the sentence "...", is *direct_object* moving, yes or no?". If a model were to answer this with "yes", we would feel confident that it has understood the CMC; however, if it answered with "no", we could not be sure that the model has failed specifically in its understanding of the CMC, and not of the sentence or situation in general. We therefore construct a control question, for which we replace the verb of the CMC with the appropriately inflected form of "throw",

[3] The specific models used were `gpt-4o-mini-2024-07-18`, `gpt-4o-2024-11-20`, `o3-mini-2025-01-31` and `o1-2024-12-17`.

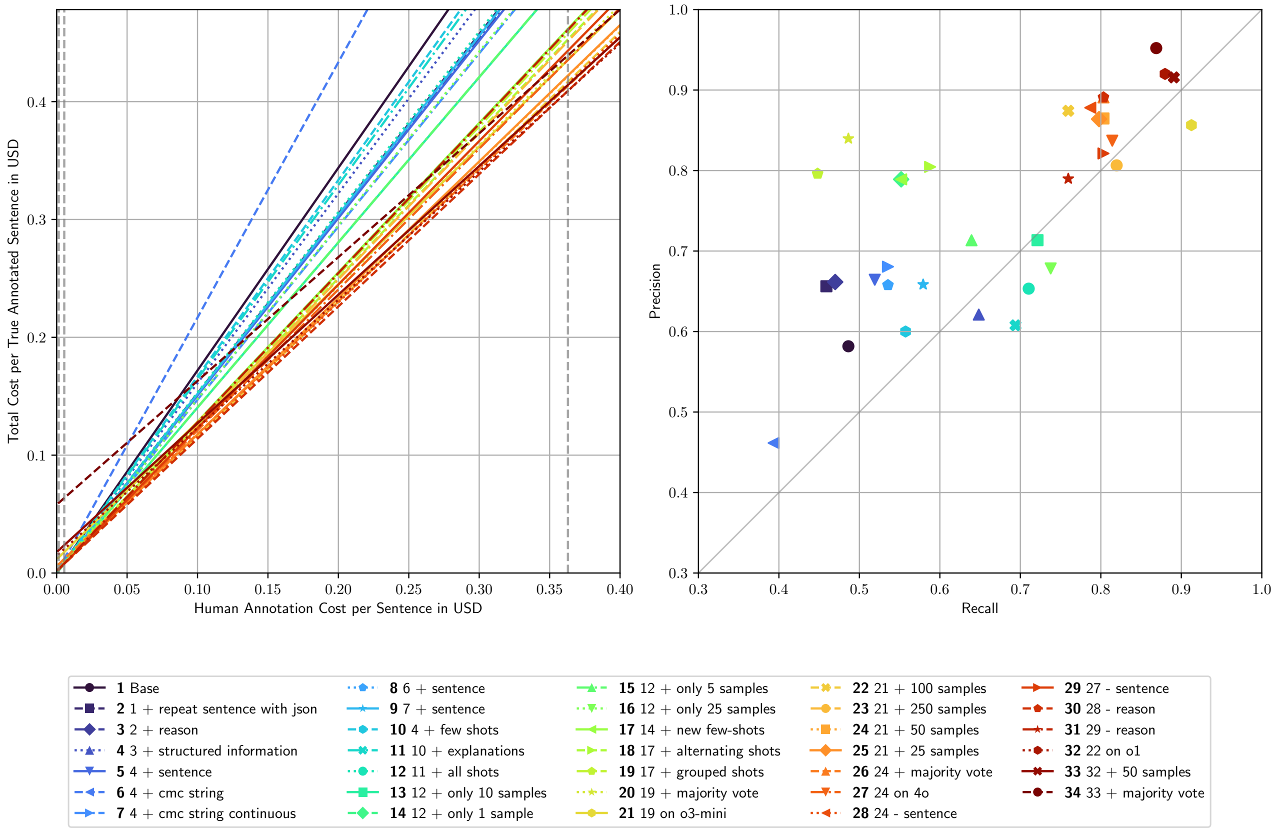


Figure 2: A comparison of all prompting setups that were considered in development. On the left, the total cost per true annotated sentence is shown dependent on the human annotation cost, in USD. On the right, prompts are compared by recall and precision.

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.

Table 3: An example prompting setup (30)

I crumble them into the bowl one at a time .
I just wept a single tear into my beard .
He hissed air through his clenched teeth .
did people really crane grand pianos to upper floors ?
Gently swirl it into the batter .

Table 4: Examples from the final dataset. Verbs are highlighted in green, direct objects in purple, prepositions in blue, and prepositional objects in red.

and ask the same question again, using the structural information extracted by the dependency filtering step. This is intended to test if the model is having a general problem understanding the sentence (which would still be an issue, but not the one we set out to find), or specifically with the CMC. While the sentence variants with “throw” are still instances of the CMC, they are now prototypical ones, which we expect to require no deeper understanding of the semantics of the CMC, as

Question Type	Example Sentence
original	In the sentence 'did people really [crane|throw] grand pianos to upper floors ?', did pianos really move, yes or no?
original_prep	In the sentence 'did people really [crane|throw] grand pianos to upper floors ?', did pianos really move to floors, yes or no?
medium	In the sentence 'People [crane|throw] grand pianos to upper floors .', do pianos move, yes or no?
medium_prep	In the sentence 'People [crane|throw] grand pianos to upper floors .', do pianos move to floors, yes or no?
short	In the sentence 'You [crane|throw] pianos to floors .', do pianos move, yes or no?
short_prep	In the sentence 'You [crane|throw] pianos to floors .', do pianos move to floors, yes or no?

Table 5: An overview of the prompt formats for LLMs, for the example sentence 'did people really crane grand pianos to upper floors?'. For each prompt, the main verb 'crane' is optionally replaced with the appropriate form of 'throw'. Each question exists once with the direct object and once without. The sentence itself is modified with two stages of simplification (medium and short).

the verb is behaving in a prototypical and frequently observed way. We expect that models with no understanding of the CMC would answer "yes" both times only for prototypical instances, and switch from "no" to "yes" for non-prototypical ones. Models with a perfect understanding of the CMC would always answer "yes".

As this only covers the most basic element of understanding the CMC sentence, the presence of motion, we also expand the evaluation paradigm to also query the destination of the caused motion. This results in a question of the format "In the sentence "...", is *direct_object* moving *prep prep_obj*, yes or no?". This is a more challenging version of the question, which will allow us to test the models on all aspects of the CMC's meaning.

Some of the sentences in our corpus contain modal verbs (e.g., *I may sneeze the foam off the cappuccino*), questions (e.g. *Did you sneeze the foam off the cappuccino?*), or other hypotheticals (e.g. *I nearly sneezed the foam off the cappuccino.*). Asking if the foam moved off the cappuccino in any of these sentences should be correctly answered with 'no', or at least with a lengthy explanation, which introduces noise into our evaluation. We therefore automatically modify each sentence using the existing dependency parse to form simpler sentences in the present tense and indicative mood, which we call "medium" sentences. In a more radical edit, we also form a "short" version, which consists only of the verb, direct object, preposition, and prepositional object, forming a sentence together with a pronoun. This is meant to evaluate if additional context helps or hinders the models in answering the question. Examples for all sentence and question types are given in Table 5.

We conduct this experiment on our corpus of 500 hand-annotated sentences. As API-based LLM, we investigate OpenAI's 4o-mini (OpenAI, 2022). From the family of open LLMs, we further choose Llama3 (Touvron et al., 2023) in sizes 8B and 70B from version 3.1, and 1B and 3B from version 3.2, Mistral 7b (Jiang et al., 2023), OLMo2 in sizes 7B and 13B (OLMo et al., 2025), Gemma3 in sizes 1B, 4B, 12B, and 27B (Team et al., 2025), as well as Aya Expanse 8B (Dang et al., 2024). Models generate a sentence in response, which we then parse for versions of "yes" and "no". We use temperature 0 for all models, i.e. greedy decoding.

4.2 Results

Figure 3 presents the results in three groups. (i) Green: the model answers "yes" both times and therefore demonstrates that it understands the CMC. (ii) Red: The model answers with "no" for the original sentence but changes its answer to "yes" when the verb is changed to "throw", meaning that it does not understand the CMC. (iii) Grey: Even with "throw", the model does not answer correctly that the direct object is moving. We consider these to be general failures of the model to understand the instruction, rather than the CMC specifically.

Indicative Present Sentences On this subgroup, titled 'medium' and 'medium_prep' in the plot, performance is higher for all models than on the questions formed with original sentences. This fits well with our intuition that the original sentences sometimes consider modals and hypotheticals, and can therefore not straightforwardly be answered with 'yes', and we therefore consider these to be the main LLM results.

Context-Free Sentences For this minimal version of the evaluation, models overall perform as well or slightly worse than for the indicative present variants. This indicates that the lack of additional context only minimally hurts model performance, and consequently, that models were only utilising the context to answer the question to a small degree.

Destination of Caused Motion If we ask only if the direct object is moving, we cannot take any model's accuracy as a direct measure of its understanding of the entire construction. It is possible that a model might understand that the direct object is moving in some way, but not precisely in which direction, and therefore wouldn't have entirely captioned the boundaries of the

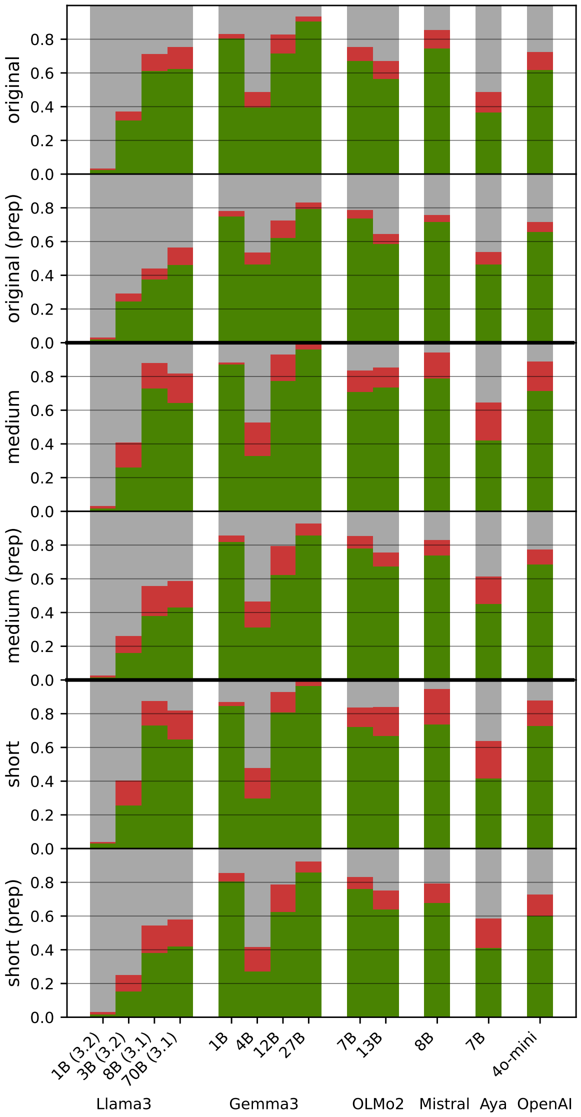


Figure 3: Results for each model and evaluation type. Examples for the evaluation types are given in Table 5. Correct answers are coloured in green, incorrect in red, and invalid results in grey.

Neither is it ever going to vibrate itself out of place .
I chop up the bacon and crumble it on top .
Do not squat the bar off the ground .
We thin the weak from the heard .
It rained arrows from the sky at any rate .

Table 6: Examples from the final dataset which were wrongly classified as negative instances by prompt 30. Verbs are highlighted in green, direct objects in purple, prepositions in blue, and prepositional objects in red.

CMC. To test this, we design a second question that includes the prepositional object, examples for which can be seen in Table 5, where the question types are suffixed with _prep.

Across the board, models give fewer correct answers to these questions than to the ones which do not include the destination (always directly above in Figure 3). However, the rate of false answers mostly stays the same or decreases, while the rate of invalid answers increases, meaning that models are more likely to answer 'no' when asked the question, including the destination of 'throw'. This may indicate that models are having general trouble interpreting these complex sentences. The pattern holds even when considering the short_prep category, where nothing else in the sentence could interfere with the model's understanding.

Results by Model Comparing different models, we find that Gemma3 perform best, with the 27B variant consistently in the range of 90%. The performance of Llama3 is correlated with model size, while that of Gemma3 is not. Gemma3 1B stands out in particular with performance almost rivalling that of the 27B version, for unknown reasons. The high performance of Gemma3 27B indicates that our questions are solvable for models, but remain a challenge for most of them. This is further supported by the fact that the only sentence types where this model falls below 90% is in the original and original_prep categories, which may include sentences where 'yes' is not the correct answer, as explained above.

4.3 Results on False Negatives

Even though our pipeline to create the test corpus included manual verification of all sentences, there is still a possibility that the automated steps introduced bias, i.e. mistakenly filtered out a set of sentences that would have significantly altered the results of our LLM evaluation. To investigate this, we repeat the same evaluation using specifically the false negatives from our corpus collection. While it would be infeasible to collect false negatives from the dependency filtering step due to the very low concentration of CMC sentences in raw data, we can take a sample of the false negatives of the LLM filtering step simply by using the false negatives from the development set that we hand-annotated earlier. With the final prompt 30, this was a set of 36 sentences that had been hand-annotated as CMC sentences, but were wrongly missed by the prompt. If the results of running the LLM evaluation on these were identical to running it on the entire collected dataset, this would tell us that the LLM filtering does not systematically exclude sentences that are more or less challenging for other LLMs to answer questions about than a random sample would have been. While we cannot find any obvious patterns in the set of false negatives, we provide some example sentences from it in Table 6.

We present the results of this in Figure 4. The results are striking: all models perform significantly worse on this set of 36 false negatives. Most interestingly, the largest change is the increase in false answers and decrease in invalid answers. This leads us to two conclusions. First, the LLMs overlap in their notion of difficulty of a CMC sentence: while the false negatives come from prompt 30, which used GPT-4o, the sentences that it misclassified were not only more difficult for 4o-mini, but also for all other models. Second, the results in the previous section, while more robust because they were based on 500, not just 36 sentences, overestimated all models' understanding of the CMC. Interestingly, the previously best model, Gemma3 27B, is now rivalled by its much smaller variant, Gemma3 1B, and neither performs as well as on the full dataset. On the other hand, specifically the short variant, which are minimal sentences where we do not ask for the destination of movement, were still almost fully solved by Gemma3 27B. It should also be noted, however, that the general relative trends between models are very similar to those of the full evaluation. This control set is, of course, also not a representation of the true distribution; it is likely that it represents exactly the most difficult subset of CMC sentences from an LLM perspective.

Overall, this has shown that while our hybrid pipeline is not perfect, the evaluation based on it still shows the general trend that most language models have large deficits in understanding the CMC, even though they are slightly underestimated.

5 Conclusion

We have introduced an annotation pipeline aided by dependency parsing and prompting LLMs, which can be specifically used for phenomena that are so rare that little to no corpora have been created, as the human annotation effort would be too great. We have demonstrated this pipeline on the example of the caused-motion construction, and a corpus of 500 caused-motion sentences.

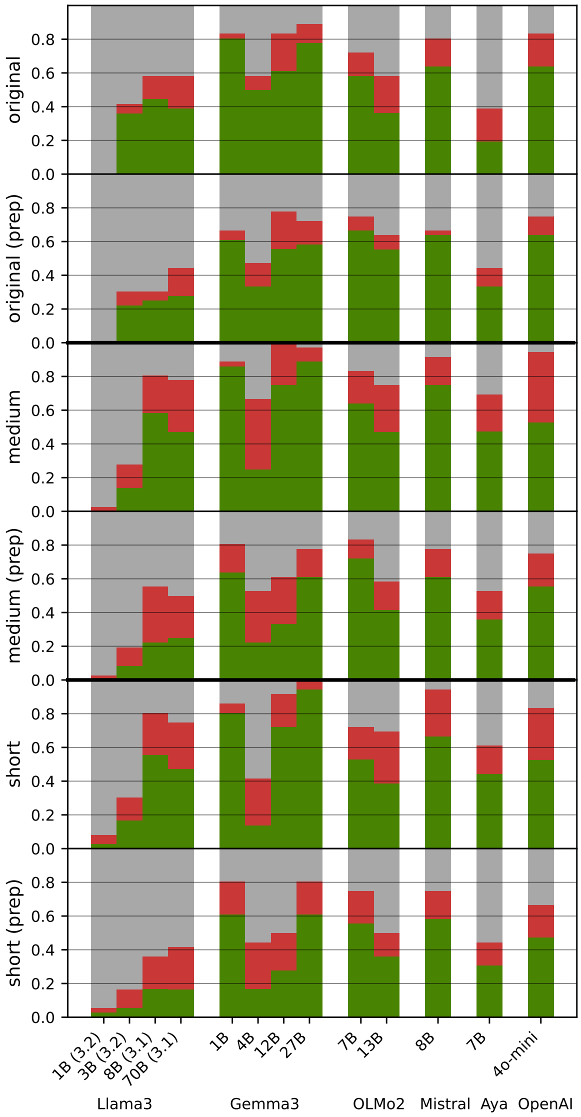


Figure 4: Results for each model and evaluation type. Examples for the evaluation types are given in Table 5. Correct answers are coloured in green, incorrect in red, and invalid results in grey.

We have used the manually annotated corpus to evaluate state-of-the-art LLMs for their understanding of the CMC, and found that many have high error rates when asked to interpret situations described with a non-prototypical CMC.

We hope that our work will inspire more computational and corpus-based studies of rare linguistic phenomena. We note that even though prompt engineering is complex, large gains can be achieved by using intermediate-complexity prompting setups and basic knowledge of LLMs. We are confident that further advances in instruction-tuned LLMs will make the cost-benefit ratio of incorporating them into this hybrid annotation pipeline even stronger.

We see several opportunities for interesting future work in both halves of the paper. For the data collection part, it is a promising engineering direction to develop tools that automate parts of this process so that it becomes available to linguists without the need for complex prompt engineering. Continued progress in LLMs is likely to make the process even more efficient.

Concerning the evaluation of LLMs' understanding of constructions, a straightforward direction for future work would be to expand to the other three Argument Structure Constructions described in Goldberg (1992).

Limitations

Due to cost reasons, the evaluation experiments were limited to replacing the verbs only with "throw". A further validation of the results could be achieve by repeating the experiment with several other prototypical motion verbs.

Because the evaluation prompts as shown in Table 5 are automatically generated, the resulting sentences might occasionally be slightly unnatural, which could affect how models reply to them.

Acknowledgements

We thank the anonymous reviewers for their thorough and constructive feedback.

References

Ahrenberg, Lars. 2007. LinES: An English-Swedish parallel treebank. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 270–273, Tartu, Estonia. University of Tartu, Estonia.

Aoyama, Tatsuya, Shabnam Behzad, Luke Gessler, Lauren Levine, Jessica Lin, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. 2023. GENTLE: A genre-diverse multilayer challenge set for English NLP and linguistic evaluation. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 166–178, Toronto, Canada. Association for Computational Linguistics.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Baumgartner, Jason, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

Behzad, Shabnam and Amir Zeldes. 2020. A cross-genre ensemble approach to robust Reddit part of speech tagging. In *Proceedings of the 12th Web as Corpus Workshop (WAC-XII)*, pages 50–56.

Bencini, Giulia ML and Adele E Goldberg. 2000. The contribution of argument structure constructions to sentence meaning. *Journal of Memory and Language*, 43(4):640–651.

Bonial, Claire and Harish Tayyar Madabushi. 2024. A construction grammar corpus of varying schematicity: A dataset for the evaluation of abstractions in language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 243–255, Torino, Italia. ELRA and ICCL.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Chomsky, Noam. 1993. *Lectures on government and binding: The Pisa lectures*. 9. Walter de Gruyter.

Croft, William. 2001. *Radical construction grammar: Syntactic theory in typological perspective*. Oxford University Press, USA.

Dang, John, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fillmore, Charles J, Russell Lee-Goldman, and Russell Rhodes. 2012. The framenet constructicon. In Hans Christian Boas and Ivan A Sag, editors, *Sign-based construction grammar*, pages 309–372. CSLI Publications Stanford.

Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Goldberg, Adele E. 1995. Constructions: A construction grammar approach to argument structure. *Chicago UP*.

Goldberg, Adele Eva. 1992. *Argument structure constructions*. University of California, Berkeley.

Gray, Morgan, Jaromir Savelka, Wesley Oliver, and Kevin Ashley. 2023. Can GPT alleviate the burden of annotation? In *Legal Knowledge and Information Systems*, pages 157–166. IOS Press.

Holter, Ole Magnus and Basil Ell. 2023. Human-machine collaborative annotation: A case study with GPT-3. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 193–206.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Hwang, Haerim and Hyunwoo Kim. 2023. Automatic analysis of constructional diversity as a predictor of efl students' writing proficiency. *Applied linguistics*, 44(1):127–147.

Hwang, Jena D. and Martha Palmer. 2015. Identification of caused motion construction. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 51–60, Denver, Colorado. Association for Computational Linguistics.

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Koptyra, Bartłomiej, Anh Ngo, Łukasz Radliński, and Jan Kocoń. 2023. Clarin-emo: Training emotion recognition models using human annotation and ChatGPT. In *International Conference on Computational Science*, pages 365–379. Springer.

Kyle, Kristopher and Hakyung Sung. 2023. An argument structure construction treebank. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 51–62, Washington, D.C. Association for Computational Linguistics.

Li, Bai, Zining Zhu, Guillaume Thomas, Frank Rudzicz, and Yang Xu. 2022. Neural reality of argument structure constructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7410–7423, Dublin, Ireland. Association for Computational Linguistics.

Mahowald, Kyle. 2023. A discerning several thousand judgments: GPT-3 rates the article + adjective + numeral + noun construction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 265–273, Dubrovnik, Croatia. Association for Computational Linguistics.

de Marneffe, Marie-Catherine, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Misra, Kanishka and Kyle Mahowald. 2024. Language models learn rare phenomena from less rare phenomena: The case of the missing AANNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 913–929, Miami, Florida, USA. Association for Computational Linguistics.

OLMo, Team, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. 2 olmo 2 furious.

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue.

Pangakis, Nicholas, Samuel Wolken, and Neil Fasching. 2023. Automated annotation with generative ai requires validation. *arXiv preprint arXiv:2306.00176*.

Rozner, Joshua, Leonie Weissweiler, Kyle Mahowald, and Cory Shain. 2025a. Constructions are revealed in word distributions.

Rozner, Joshua, Leonie Weissweiler, and Cory Shain. 2025b. Babylm's first constructions: Causal interventions provide a signal of learning.

Sanguinetti, Manuela and Cristina Bosco. 2015. Parttut: The turin university parallel treebank. In Roberto Basili, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti, and Maria Simi, editors, *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*, pages 51–69. Springer International Publishing, Cham.

Savelka, Jaromir and Kevin D Ashley. 2023. The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts. *Frontiers in Artificial Intelligence*, 6.

Scivetti, Wesley, Tatsuya Aoyama, Ethan Wilcox, and Nathan Schneider. 2025. Unpacking let alone: Human-scale models generalize to a rare construction in form but not meaning.

Silveira, Natalia, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tayyar Madabushi, Harish, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets construction grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Team, Gemma, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 technical report.

Torrent, Tiago Timponi, Thomas Hoffmann, Arthur Lorenzi Almeida, and Mark Turner. 2023. *Copilots for Linguists: AI, Constructions, and Frames*. Cambridge University Press.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. Preprint, arXiv 2302.13971.

Tseng, Yu-Hsiang, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. CxLM: A construction and context-aware language model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6361–6369, Marseille, France. European Language Resources Association.

Warstadt, Alex, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Warstadt, Alex, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Weissweiler, Leonie, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your syntax, the better your semantics? probing pretrained language models for the English comparative correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yu, Danni, Luyang Li, Hang Su, and Matteo Fuoli. 2023. Assessing the potential of llm-assisted annotation for corpus-based pragmatics and discourse analysis: The case of apologies. *International Journal of Corpus Linguistics*.

Zeldes, Amir. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Zeman, Daniel, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Zhou, Shijia, Leonie Weissweiler, Taiqi He, Hinrich Schütze, David R. Mortensen, and Lori Levin. 2024. Constructions are so difficult that Even large language models get them right for the wrong reasons. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3804–3811, Torino, Italia. ELRA and ICCL.

A Full details for each prompt

We report in Tables 7 to 24 the details of the prompt, along with the change that it represents from a previous prompt.

B Few Shots

In Table 41, we give the five shots from each class given to ChatGPT as examples.

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Reply with a csv codeblock (wrapped in three backticks), with the headers ’id’ and ’label’. label should be either True or False. Label all 50 sentences.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	Base
Shot Strategy	all

Table 7: Prompt 1

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	1 + repeat sentence with json
Shot Strategy	all

Table 8: Prompt 2

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	2 + reason
Shot Strategy	all

Table 9: Prompt 3

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "…", "sentence": "…" }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	3 + structured information
Shot Strategy	all

Table 10: Prompt 4

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "…", "sentence": "…" }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	4 + sentence
Shot Strategy	all

Table 11: Prompt 5

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "…", "sentence": "…" }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	4 + cmc string
Shot Strategy	all

Table 12: Prompt 6

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	4 + cmc string continuous
Shot Strategy	all

Table 13: Prompt 7

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	6 + sentence
Shot Strategy	all

Table 14: Prompt 8

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	7 + sentence
Shot Strategy	all

Table 15: Prompt 9

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 10 positive examples: . Here are 10 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	10
Sentences	50
Model	4o_mini
Majority Vote	No
Change	4 + few shots
Shot Strategy	first of each verb and class

Table 16: Prompt 10

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 10 positive examples: . Here are 10 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	10
Sentences	50
Model	4o_mini
Majority Vote	No
Change	10 + explanations
Shot Strategy	first of each verb and class

Table 17: Prompt 11

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	4o_mini
Majority Vote	No
Change	11 + all shots
Shot Strategy	all

Table 18: Prompt 12

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	10
Model	4o_mini
Majority Vote	No
Change	12 + only 10 samples
Shot Strategy	all

Table 19: Prompt 13

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	No
Change	12 + only 1 sample
Shot Strategy	all

Table 20: Prompt 14

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	5
Model	4o_mini
Majority Vote	No
Change	12 + only 5 samples
Shot Strategy	all

Table 21: Prompt 15

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "…", "sentence": "…" }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	25
Model	4o_mini
Majority Vote	No
Change	12 + only 25 samples
Shot Strategy	all

Table 22: Prompt 16

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "…", "sentence": "…" }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	No
Change	14 + new few-shots
Shot Strategy	all

Table 23: Prompt 17

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "…", "sentence": "…" }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	No
Change	17 + alternating shots
Shot Strategy	all_alternating

Table 24: Prompt 18

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	No
Change	17 + grouped shots
Shot Strategy	all_grouped

Table 25: Prompt 19

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	Yes
Change	19 + majority vote
Shot Strategy	all_grouped

Table 26: Prompt 20

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	o3_mini
Majority Vote	No
Change	19 on o3-mini
Shot Strategy	all_grouped

Table 27: Prompt 21

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	100
Model	o3_mini
Majority Vote	No
Change	21 + 100 samples
Shot Strategy	all_grouped

Table 28: Prompt 22

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	250
Model	o3_mini
Majority Vote	No
Change	21 + 250 samples
Shot Strategy	all_grouped

Table 29: Prompt 23

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	o3_mini
Majority Vote	No
Change	21 + 50 samples
Shot Strategy	all_grouped

Table 30: Prompt 24

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 25 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 25 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	25
Sentences	50
Model	o3_mini
Majority Vote	No
Change	21 + 25 samples
Shot Strategy	all_grouped

Table 31: Prompt 25

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	o3_mini
Majority Vote	Yes
Change	24 + majority vote
Shot Strategy	all_grouped

Table 32: Prompt 26

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	4o
Majority Vote	No
Change	24 on 4o
Shot Strategy	all_grouped

Table 33: Prompt 27

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	o3_mini
Majority Vote	No
Change	24 - sentence
Shot Strategy	all_grouped

Table 34: Prompt 28

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	4o
Majority Vote	No
Change	27 - sentence
Shot Strategy	all_grouped

Table 35: Prompt 29

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	o3_mini
Majority Vote	No
Change	28 - reason
Shot Strategy	all_grouped

Table 36: Prompt 30

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	4o
Majority Vote	No
Change	29 - reason
Shot Strategy	all_grouped

Table 37: Prompt 31

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes.
Few-Shots	50
Sentences	100
Model	o1
Majority Vote	No
Change	22 on o1
Shot Strategy	all_grouped

Table 38: Prompt 32

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes.
Few-Shots	50
Sentences	50
Model	o1
Majority Vote	No
Change	32 + 50 samples
Shot Strategy	all_grouped

Table 39: Prompt 33

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes.
Few-Shots	50
Sentences	50
Model	o1
Majority Vote	Yes
Change	33 + majority vote
Shot Strategy	all_grouped

Table 40: Prompt 34

Sentence	Verb	Dir Obj	Prep	P-Obj	Lab.	Explanation
I actually giggled myself to tears .	giggle	myself	to	tear	False	This is a negative example because being 'in tears' is a state, not a location, so the subject here didn't move but rather changed state.
Nope , they just giggle their microscopic excretions into the air .	giggle	excretion	into	air	True	This is a positive example because the act of giggling is causing the excretions to move
I 'll stop it from repeating and fade it into a single background color .	fade	it	into	color	False	This is a negative example which describes the act of fading a color so that it can't be told apart from the background color, which means that nothing moved.
Just hover your mouse over it	hover	mouse	over	it	False	This is a negative example because the mouse is hovering over it, but it is not moving, it is staying in place while hovering.
Once she was strapped back in he started to hover her out of the room .	hover	she	out of	room	True	This is a positive example because they are moving her out of the room by hovering.
They tug him to the ground and start jumping on him and licking his face .	tug	he	to	ground	True	This is a positive example because someone was tugged and that moved him to the ground.
I gulped it from the bottle while watching old movies .	gulp	it	from	bottle	True	This is a positive example because what was in the bottle moved from the bottle because the person was drinking it.
I would rather take the spoon , I can gulp it in one go .	gulp	it	in	go	False	This is a negative example because one go is not a destination, it specifies the manner of gulping.
Cruz was trailing Clinton in basically every poll .	trail	Clinton	in	poll	False	This is a negative example because no movement is happening, the sentence describes the relative position of two politicians in a poll.
She began trailing a finger down his chest .	trail	finger	down	chest	True	This is a positive example because she is moving the finger down his chest.
He stopped for ten minutes while wheezing himself to death .	wheeze	himself	to	death	False	This is a negative example because death is a state, not a physical location.
It is not cute to watch your dog wheeze himself to the floor because he was so excited you picked up his tug of war rope .	wheeze	himself	to	floor	True	This is a positive example because the wheezing is causing the dog to move to the floor.
I bounced it off the wall .	bounce	it	off	wall	True	This is a positive example because the ball moved off the wall.
We bounce ideas off each other .	bounce	idea	off	other	False	This is a negative example because an idea can't physically move.
I was in bed for about a week and thought I was going to shiver myself to death .	shiver	myself	to	death	False	This is a negative example because death is a state, not a destination of a physical movement.
She needs to stop darting her eyes to the side every time she says something	dart	eye	to	side	False	This is a negative example because her eyes are rotating but they're not moving.
He nervously darted his tongue into her mouth .	dart	tongue	into	mouth	True	This is a positive example because his tongue is moving into her mouth.
For some reason every time i overflow the sink in Dalia 's bathroom , the Sheik always comes up to investigate ...	overflow	sink	in	bathroom	False	This is a negative example because the sink is not moving.
Most importantly the toilet was overflowing water into the pan , almost on constant flush .	overflow	water	into	pan	True	This is a positive example because the water is moving into the pan.
You 're trying to wriggle your way out of it now !	wriggle	way	out of	it	False	This is a negative example because while something is moving, it is not the direct object way.
At one point he wriggles himself into position to block a soccer ball with his head while Latin on the street .	wriggle	himself	into	position	True	This is a positive example because he is moving himself into position.
I swim laps in the pool .	swim	lap	in	pool	False	This is a negative example because while I am moving, the laps are not moving.
My wife lapped me on the scoring track .	lap	I	on	track	False	This is a negative example because I am moving, but my wife is not causing me to move.
He will nibble you to death !	nibble	you	to	death	False	This is a negative example because death is a state, not a location.
I eat my Twix by nibbling the chocolate off the sides , then off the top , then eat the caramel and cookie .	nibble	chocolate	off	side	True	This is a positive example because the chocolate is moving off the sides.
I aimed at her , and gazed her in her eyes before I successfully hit her face with the snowball .	gaze	she	in	eye	False	This is a negative example because a gaze is not something that can physically move.
I choose to be the one that goes hiking with friends into waterfalls , out galloping horses in open fields , and having fun times with my SO .	gallop	horse	in	field	False	This is a negative example because the horses are moving, but they are not moving in the direction of the field, they are already in it.
I can confirm that galloping a horse through an open field is amazing .	gallop	horse	through	field	True	This is a positive example because the horse is moving through the field.
I scramble them in the hot pan .	scramble	they	in	pan	False	This is a negative example because the eggs are not moving in the direction of the pan, they are already in it.
Once it firms a little , scramble it into the rice .	scramble	it	into	rice	True	This is a positive example because the eggs are moving into the rice.
To continue with your explanation , we see not only that this man here can afford to encrust rare and obviously expensive jewels onto his box of ' Fruity Pebbles ' brand breakfast cereal , but also that he can afford the ' Family Size ' box .	encrust	jewel	onto	box	True	This is a positive example because the jewels are moving onto the box.
I peel paint off walls .	peel	paint	off	wall	True	This is a positive example because the paint is moving off the wall.
I peel bananas from the bottom	peel	banana	from	bottom	False	This is a negative example because the banana is not moving, only the peel is, and it is not moving from the bottom.
In my defense I was actually very drunk when I plowed my car into that crowd of pedestrians .	plow	car	into	crowd	True	This is a positive example because I caused tha car to move into the crowd.
I plow snow in the winter	plow	snow	in	winter	False	This is a negative example because in the winter is a time, not a location.
And drag queens cake themselves in makeup .	cake	themselves	in	makeup	False	This is a negative example because the drag queens are not moving.
I would cake makeup on my face to hide it .	cake	makeup	on	face	True	This is a positive example because the makeup is moving onto the face.
Whereas WWE charred it to a crisp and drowned it in A-1 sauce .	char	it	to	crisp	False	This is a negative example because it is changing state to a crips, not moving.
I fermented it in a 3 gallon food grade plastic bucket .	ferment	it	in	bucket	False	This is a negative example because it is staying in the bucket and not moving.
When the child collapsed , the mother hurried him to the hospital , where he died .	hurry	he	to	hospital	True	This is a positive example because the child is moving to the hospital.
I will take my time or hurry you through a meal , there are no rules against that .	hurry	you	through	meal	False	This is a negative example because the meal here is an action, not a destination
I love blackening it in a roasting pan .	blacken	it	in	pan	False	This is a negative example because it is not moving, it is staying in the pan.
I rarely use them but my girlfriend is crocheting them into reusable shopping bags ...	crochet	they	into	bag	False	This is a negative example because the bags are not moving, they are being made into something else.
When " nice guys " change their MO to target " nice girls " the equilibrium will tilt the earth off its axis and hurtle us into space , thus settling this tired argument for all eternity .	hurtle	we	into	space	True	This is a positive example because we are moving into space.
Then you drip juice into it and vape .	drip	juice	into	it	True	This is a positive example because the juice is moving into it.
As in you literally gnaw it off the bone .	gnaw	it	off	bone	True	This is a positive example because it is moving off the bone
I twitch my head to the side .	twitch	head	to	side	True	This is a positive example because the head is moving to the side.
He snorted coke off my ass	snort	coke	off	ass	True	This is a positive example because the coke moved off my ass.
I ca n't tell if she 's smiling or is she 's about to sneeze the sand off of her nose .	sneeze	sand	off of	nose	True	This is a positive example because the sand moves off her nose.
It was like a little rocket that tried to burrow itself into the ground .	burrow	itself	into	ground	True	This is a positive example because the rocket moves into the ground.

Table 41: Few shots. P-Obj stands for Prepositional Object, Dir Obj for Direct Object.

Part III

Morphology

Chapter 10

Corresponds to the following publication:

> **Leonie Weissweiler***, Valentin Hofmann*, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, David R. Mortensen (2023). Counting the Bugs in ChatGPT's Wugs: A Multilingual Investigation into the Morphological Capabilities of a Large Language Model. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. (EMNLP) *Equal contribution.

Declaration of Co-Authorship David Mortensen conceived the initial research contribution. Valentin Hofmann and Kemal Oflazer helped develop the details of the morphological research question. Anjali Kantharuban worked on Tamil and Kemal Oflazer on Turkish, each gathering human data, translating the English prompt setup and giving feedback on the results. Anna Cai, Atharva Kulkarni, Abhishek Vijayakumar, Ritam Dutt and Haofei Yu each implemented one of the baselines. Amey Hengle and Anubha Kabra conducted initial prompting experiments, which were further developed and executed by Valentin Hofmann and myself. I coordinated the efforts and explained the inputs, outputs and context in the project to each author. I performed the evaluation of all baselines and of ChatGPT and generated figures. All authors contributed parts of the initial draft, and I led the completion of the paper in close collaboration with David Mortensen, Valentin Hofmann and Kemal Oflazer.

Research Context This part of the thesis tackles similar questions to the CxG part. Morphology presents unique opportunities to extend this work, because it is considered to be less regular than Syntax. It is also underrepresented in NLP: when preparing this work, we found that there was no prior work evaluating any LLM for its morphological capabilities systematically. With the motivation that in the current climate, any possible capability of LLMs that is understudied may lead to overstated claims of human-level abilities, we therefore adapt the classic wug-test for evaluating them.

Counting the Bugs in ChatGPT's Wugs: A Multilingual Investigation into the Morphological Capabilities of a Large Language Model

Leonie Weissweiler[*2,4], Valentin Hofmann[*2-5], Anjali Kantharuban[1], Anna Cai[†1], Ritam Dutt [†1], Amey Hengle[†6], Anubha Kabra[†1], Atharva Kulkarni[†1], Abhishek Vijayakumar[†1], Haofei Yu[†1], Hinrich Schütze[2,4], Kemal Oflazer[1], David R. Mortensen[1]

[1]Carnegie Mellon University [2]LMU Munich [3]University of Oxford
[4]Munich Center for Machine Learning [5]Allen Institute for AI [6]IIT Delhi
weissweiler@cis.lmu.de

Abstract

Large language models (LLMs) have recently reached an impressive level of linguistic capability, prompting comparisons with human language skills. However, there have been relatively few systematic inquiries into the linguistic capabilities of the latest generation of LLMs, and those studies that do exist (i) ignore the remarkable ability of humans to generalize, (ii) focus only on English, and (iii) investigate syntax or semantics and overlook other capabilities that lie at the heart of human language, like morphology. Here, we close these gaps by conducting the first rigorous analysis of the morphological capabilities of ChatGPT in four typologically varied languages (specifically, English, German, Tamil, and Turkish). We apply a version of Berko's (1958) wug test to ChatGPT, using novel, uncontaminated datasets for the four examined languages. We find that ChatGPT massively underperforms purpose-built systems, particularly in English. Overall, our results—through the lens of morphology—cast a new light on the linguistic capabilities of ChatGPT, suggesting that claims of human-like language skills are premature and misleading.

1 Introduction

Do large language models (LLMs) possess human-like linguistic capabilities? With the advent of the latest generation of LLMs such as GPT-4 (OpenAI, 2023b), LLaMA (Touvron et al., 2023), and PaLM (Chowdhery et al., 2022), there appears to be growing evidence for answering this question with *yes* (Bubeck et al., 2023): LLMs are capable of generating text that crowdworkers cannot distinguish from human-generated text (Clark et al., 2021) and excel at linguistic probing tasks such as predicting grammaticality, detecting the subject and tense of clauses, and identifying the grammatical number of subjects and objects (Jin et al., 2022).

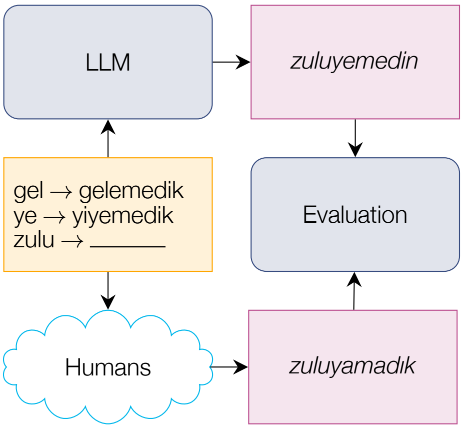


Figure 1: Experimental paradigm for this study (illustrated with Turkish). Human annotators and an LLM are given examples and a nonce word to be inflected. The generated inflected forms are compared.

Despite these encouraging results, the existing body of work has so far examined a relatively limited part of the full spectrum of phenomena that are known to characterize human language, with a heavy focus on syntax and semantics. One area that has been neglected in particular is *morphology*, i.e., the capacity to create words according to systematic patterns of covariation in form and meaning (Haspelmath and Sims, 2010). This gap in the LLM literature is noteworthy given that morphology has been a hallmark of research on computational approaches to language since the very beginnings of neural language processing in the 1980s (Rumelhart and McClelland, 1986b; Plunkett and Juola, 1999; Albright and Hayes, 2002, 2003; Goldberg, 2019).

In this study, we present the first systematic analysis of the morphological capabilities of LLMs, fo-

[*]Equal contribution.
[†]Authors sorted alphabetically.

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 6508–6524
December 6-10, 2023 ©2023 Association for Computational Linguistics

cusing on ChatGPT (OpenAI, 2023a) as the most prominent and most widely-used LLM. Specifically, we investigate ChatGPT's morphological capabilities using the wug test (Berko, 1958), an experimental paradigm in which a participant is asked to provide an inflected or derived form of a nonce word. An example for our evaluation setup is given in Figure 1. Our experiments cover a broad range of morphological constructions and four typologically diverse languages: English, German, Tamil, and Turkish. We find that ChatGPT falls short not only of human performance but also of various supervised baselines.

In sum, our contributions are as follows:

- We conduct the first systematic analysis into the morphological capabilities of LLMs.

- Our study covers a diverse set of morphological constructions/languages and introduces datasets for future research in the area.[1]

- We show that ChatGPT has not achieved human parity—or even state-of-the-art performance—on our nonce-word inflection/reinflection tasks but performs about as well as some older supervised models. We furthermore find evidence for the existence of a real word bias in ChatGPT that is the more pronounced the more data ChatGPT has seen for a given language.

2 Related Work

2.1 Computational Morphology

Linguists divide morphology into inflection and derivation (Haspelmath and Sims, 2010). While inflection accounts for the different word forms of a lexeme, e.g., *listen*, *listens*, and *listened*, derivation accounts for the different lexemes of a word family, e.g., *listen*, *listener*, and *listenable*. Both inflection and derivation have been addressed in computational linguistics and natural language processing (NLP), albeit with a heavy focus on inflection. One line of work, which is conceptually similar to wug testing, has sought to generate inflected forms, given a stem and a morphological tag (Cotterell et al., 2017a, 2018; Vylomova et al., 2020; Goldman et al., 2022), using systems ranging from weighted finite state transducers and GRU/LSTM encoder-decoder models (with soft attention or hard monotonic attention) to various transformer models. A special subtype of this task is morphological reinflection, where the input can be a form that is itself inflected (Cotterell et al., 2016a; Kann and Schütze, 2016; Kann et al., 2017; Silfverberg et al., 2017; Pimentel et al., 2021). Other typical tasks in computational research on inflection are morphological segmentation (Cotterell et al., 2015, 2016b,c; Kann et al., 2016), unsupervised morphology induction (Hammarström and Borin, 2011; Soricut and Och, 2015; Xu et al., 2018; Weissweiler et al., 2022), and morphological paradigm completion (Erdmann et al., 2020a,b; Jin et al., 2020). There has also been some interest in the modeling of derivation (Cotterell et al., 2017b; Vylomova et al., 2017; Deutsch et al., 2018; Hofmann et al., 2020b,c).

More recently, there have been a few studies examining the morphological capabilities of language models (Edmiston, 2020; Hofmann et al., 2020a), but they focus on smaller language models such as BERT (Devlin et al., 2019). By contrast, we examine ChatGPT, a model whose parameter count is three orders of magnitude larger, and we analyze its zero-, one-, and few-shot capabilities, an approach fully neglected by prior work.

2.2 Multilingual Capabilities of LLMs

Recent studies have extensively examined the evaluation of LLMs in multilingual settings. Some of these studies have specifically investigated the extent to which LLMs can be used for traditional multilingual NLP tasks such as machine translation (Bawden et al., 2022; Hendy et al., 2023; Jiao et al., 2023; Wang et al., 2023). Brown et al. (2023) demonstrate that LLMs perform well across multiple languages even with minimal task-specific training, highlighting their transferability and generalization in multilingual understanding.

2.3 LLM Performance on Unseen Data

The fact that LLMs have been pretrained on massive amounts of data means that they have seen and potentially memorized a substantial amount of the items of data used in typical evaluation setups (Magar and Schwartz, 2022). There have been a few attempts in NLP to specifically control for previous exposure (Haley, 2020; Hofmann et al., 2020a; Maudslay and Cotterell, 2021). We follow this idea by generating datasets of novel and uncontaminated nonce words, thus ensuring that the words have not been seen by ChatGPT before.

[1] We release our dataset along with our code at https://github.com/dmort27/chatgpts-wugs, carefully following the guidelines laid out by Jacovi et al. (2023).

3 Data and Morphological Constructions

In this paper, we examine ChatGPT's morphological behavior on a typologically diverse set of languages: English, German, Tamil, and Turkish. While English and German belong to the same language family, German has a more fusional morphological system than English. Turkish is chosen since it is a non-Indo-European language with a fully agglutinative morphology. Tamil is chosen since it is a Dravidian language exhibiting an agglutinative morphology with fusional elements. Thus, in terms of the classical triangle of fusional, isolating, and agglutinative morphologies (Dixon, 1994), the languages cover four different points: almost fully isolating (English), intermediate between isolating and fusional (German), intermediate between fusional and agglutinative (Tamil), and fully agglutinative (Turkish). Furthermore, the chosen languages also cover different points in the spectrum from low-resource to high-resource, enabling us to form hypotheses about the impact of the amount of language-specific training data on the morphological capabilities of an LLM. Statistics for the amount of data in train, dev, and test for the baselines, as well as the number of wug test words, are given in Table 1. We report the accuracy of one annotator at a time against the judgments of all other annotators in Table 2.

Lang.	Train	Dev	Test	Wug test
English	10,000	1,000	1,000	50
German	10,000	1,000	1,000	174
Tamil	1,541	368	—	123
Turkish	8,579	851	846	40

Table 1: Data statistics. Please see Appendix A.1 for the distribution of morphological tags across the different splits for the four languages. There was not enough data available for Tamil to form a test set.

	Accuracy (%)		
Lang.	@1	@3	@5
English	67.14 ± 17.76	85.29 ± 13.06	87.64 ± 12.13
German	63.05 ± 12.62	83.80 ± 10.57	87.88 ± 10.34
Tamil	37.09 ± 26.39	43.85 ± 26.95	43.85 ± 26.95

Table 2: Accuracy of one annotator at a time against the judgments of the other annotators on our collected wug dataset, for different values of k. For Turkish, since the morphology is deterministic, there is no variation.

3.1 English

The English past tense has a long and storied history in computational studies of morphology (Rumelhart and McClelland, 1986a; Pinker and Prince, 1988; Ullman et al., 1997; Plunkett and Juola, 1999; Albright and Hayes, 2002, 2003; Kirov and Cotterell, 2018; Ma and Gao, 2022). English displays a handful of conjugation classes as well as frequent morphographemic alternations—consonant doubling and e-deletion, for example—affecting past forms of verbs.

To create the English data, 50 two- to five-letter irregular verbs (defined as verbs that do not form the past tense simply by adding *-ed*) were sampled from the UniMorph 4.0 dataset (Batsuren et al., 2022). These items were each perturbed by one or two letters (substituting phonetically similar sounds) producing a word not included in UniMorph. These verbs were then annotated by 28 volunteer annotators. Participants were asked to provide the past tense of the nonce word and given an example (*wug* → *wugged*) and the frame "They {nonce_word} all the time. In fact, they ____ just yesterday." This yielded mappings between a lemma and a ranked list of inflected verbs, e.g., *veed* → [*veeded*, *ved*, *vode*]. The modal annotation was always a regularly inflected form (*-ed* with appropriate allomorphic variation), but other inflectional classes were attested.

3.2 German

The German plural of nouns is a morphological phenomenon intensely studied in linguistics and the cognitive sciences due to the general complexity of the alternation between the eight different operations that can be used to express it. German pluralization is particularly notable due to the fact that none of the possible operations express it in a majority of cases (McCurdy et al., 2020). In fact, the most frequent German plural noun suffix *-en* has been argued not to be the default (i.e., the suffix that applies to novel nouns)—an honor that goes to *-s* (Marcus et al., 1995).

To create the dataset of novel German nonce nouns, we drew upon Unipseudo.[2] We generated 200 nonce words with a length between four and seven characters (50 nonce words per character length), using German nouns as input to the algorithm. We then had one German native speaker unrelated to the study (i) generate articles (*der*, *die*, or *das*) for each of the nonce words, and (ii) generate a plural based on the nonce words and the previ-

[2] http://www.lexique.org/shiny/unipseudo/

ously selected articles. We manually filtered out words whose plural is blocked by existing German lexemes, resulting in a final set of 174 nonce nouns. These nouns were then annotated by 21 volunteer annotators. Participants were asked to provide the plural of the nonce word and were given an example (*Wug* → *Wugs*) and the frame "Hier ist ein/e {nonce_word}. Jetzt sind es zwei ____." Similarly to English, this yielded mappings between a lemma and a ranked list of inflected nouns.

Features	Example
First person singular agreement and past tense	zöbür-ür-üm → zöbür-dü-m
Second person plural agreement, reported/inferential past tense, and negative polarity	zöbür-ür-sünüz → zöbür-me-miş-siniz
Dative case, first person possessive	zürp-ten → zürb-üm-e
Accusative singular	börüt → börüd-ü

Table 3: Turkish tasks. Forms with colored suffixes are actually used in the long prompt in a contextually meaningful short sentence. Hyphens represent morpheme boundaries. The last row is for simple inflection. The predicted forms (to be predicted, on the right) have the following morphosemantics: "I [verb]-ed", "(I heard that) you have not [verb]-ed", "to my [noun]", "the [noun] (as a definite object)".

3.3 Tamil

Tamil is a Dravidian language primarily spoken in regions of South India and Sri Lanka. It is an agglutinative languange in which verbs are conjugated for tense, transitivity, person, number, and (in some cases) gender. For the most part, affixes display allomorphy only due to phonological conditioning and are otherwise invariant across verbs, as is the case with the person/number/gender (PNG) affix (Arden, 1891, 71). This is not the case, however, for tense markers. Among linguists working on Tamil, it is not completely agreed upon how many verb classes there are in the language, with some proposing up to 13 and others as few as three (Lisker, 1951; Agesthialingom, 1971). In the spoken form of Tamil, there are points where verbs are part of completely different classes than their literary counterpart, so in this study we focus exclusively on the written form (Schiffman and Renganathan, 2009).

To simplify the analysis, we utilize a modification of Graul's classification seen in *The English Dictionary of the Tamil Verb*, where there are seven primary classes (Schiffman and Renganathan, 2009). The tense most impacted by these verb classes is the past tense, with each class having a unique form, while the present and future only demonstrate three forms across the classes. As such, we focus on the past tense and designate the same transitivity (intransitive) and PNG (third person singular masculine) affix across all experiments. In examining this, we gain information about the ways LLMs handle morphologically complex languages with inflectional classes defined in both phonological and morphological terms. This contrasts with English, where inflection is not agglutinative, and Turkish, where morphology is agglutinative but where there are no inflectional classes.

To create a dataset for training the baseline models and generating samples for the few-shot prompts, 86 common Tamil verbs were sampled and conjugated with every possible combination of tense and PNG suffixes. These conjugations were generated automatically and then validated by two native speakers for accuracy. Unlike in the nonce word case, there was 100% agreement between speakers. The nonce words were generated by combining syllables from real verb roots and checking against a Tamil dictionary to assure the words created were not real. Nonce verbs were created to be between two and six letters long to best match the distribution of real Tamil verbs. In order to get the "correct" past tense for these verbs, five native Tamil speakers were asked to provide past tense forms (e.g., நிடு *niṭu* → [நிடுத்தான் *niṭuṯ:a:n*, நிட்டான் *niṭ:a:n*, நீடினான் *ni:ṭina:n*]). The mode of these responses was taken to be the gold form, with the level of agreement amongst speakers recorded for later analysis. The comparatively lower inter-annotator agreement can be explained by the lack of historical and linguistic context given to the annotators, since a large part of classification is historical.

3.4 Turkish

Turkish is an agglutinative language where words consist of multiple morphemes attached to a root. Surface realizations of morphemes are influenced by deterministic morphophonological processes like vowel harmony, consonant assimilation, and elision. Unlike many other languages, Turkish has complex word form morphotactics, particularly when multiple derivations are present.

To simplify the task and reduce the number of feature combinations, we utilized four datasets with different levels of complexity and a limited number of inflectional features. In most cases, the context provides an inflected form with one set of features, and the model must predict the form with the requested set of features. The first three tasks are reinflection tasks, demanding proficiency in both morphotactics and morphographemics. The fourth task is a straightforward inflection task (see Table 3). Each task consists of up to five shot examples for real roots and 10 test examples with nonce roots. Stimuli and gold annotations were produced by our (single) Turkish annotator.

4 Methodology

We compare the outputs of ChatGPT under a variety of prompting regimens and a substantial set of supervised baselines (both neural and non-neural) to human annotations of the data described in Appendix 3. Results are evaluated using accuracy at *k* (*acc@k*), i.e., a model's response is regarded as correct if it is in line with any of the top *k* human responses. This evaluation method takes into account inter-speaker morphological variability, which is more wide-spread than previously thought (Dammel and Schallert, 2019).

4.1 Baselines

We investigate the efficacy of several baselines for the task of morphological inflection. The chosen baselines encompass both statistical and neural architectures that have shown impressive performance on the morphological generalization task in recent years. We evaluate their performance on the SIGMORPHON 2023 task as well as on our constructed wug test set. The baselines have complementary strengths (see Section 5).

4.1.1 Training Data

We used the train/dev/test splits of the SIGMORPHON 2023 Inflection Shared Task[3] for English and German. The choice of the train/dev/test splits was motivated by the fact that there was no overlap of lemmata between the individual splits, thus mimicking a wug-like setting.

The Turkish training data for baselines was generated directly using a Turkish morphological analyzer/generator (Oflazer, 1994), because the aforementioned SIGMORPHON 2023 dataset did not have a sufficient number of examples for most of the feature combinations. The morphological generator was set up to generate only Turkish word forms that corresponded to the selected inflectional morpheme combinations we selected, for *all* applicable roots. For testing, we expected the baseline systems to generate the word forms with the selected inflectional feature combinations, but *for 10 nonce roots*. The nonce roots were chosen so that they would force the inflected forms to orthogonally adhere to surface morphographemic constraints and rules such as various types of vowel harmony, consonant elision, or assimilation at morpheme boundaries.

Similarly, for Tamil, we split the data into train and dev sets. Since we have a limited amount of Tamil data, we kept the split ratio at around 4:1 between train and dev sets.

We report the results of all baselines in Table 4. Baselines generally perform as expected, validating our usage of them. It should be noted that MinGen and AED are evaluated in IPA/feature space and may therefore be at a disadvantage compared to baselines operating directly in orthography. The training data was converted from orthography into IPA using Epitran (Mortensen et al., 2018).

4.1.2 Affix Rule Learner (ARL)

As a baseline for the 2020 and 2021 SIGMORPHON shared tasks, a simple non-neural system (Liu and Mao, 2016) was implemented that uses edit distance to "discover prefix and suffix rules in training data."[4] At test time, the system modifies a lemma by applying the longest matching suffix rule and most frequently applied prefix rule for a given morphosyntactic description.

4.1.3 Minimal Generalization Learner (MinGen)

Wilson and Li (2021) proposed a minimal generalization model based on a simplified form of Albright and Hayes (2002) to learn morphological rules. First, base rules that describe the changes needed to convert a lemma to an inflected form are generated from training data. The rules are further generalized by comparing phonological features of the rule contexts. The rules are then scored by a confidence metric based on their accuracy and

[3] https://github.com/sigmorphon/2023InflectionST

[4] https://github.com/sigmorphon/2021Task0/tree/main/baselines

	English		German		Turkish		Tamil
Model	Dev	Test	Dev	Test	Dev	Test	Dev
ARL	95.40	96.60	77.40	79.80	94.36	93.50	85.60
MinGen	81.40	78.70	72.70	70.70	93.65	93.03	87.23
FIT	96.22 ± 0.19	94.93 ± 0.49	79.01 ± 1.16	81.04 ± 1.39	97.00 ± 0.22	96.25 ± 0.26	64.24 ± 3.11
PPI	95.95 ± 0.63	94.74 ± 0.90	73.57 ± 5.37	78.26 ± 4.66	96.61 ± 0.60	96.56 ±0.66	76.76± 2.10
AED	71.06 ± 5.74	70.16 ± 5.79	64.44 ± 1.85	67.44 ± 2.02	95.54 ± 0.77	95.19 ±1.41	50.70± 2.84

Table 4: Results (*acc@k*) of the baselines on our development and test data. See Section 4.1.1 for full details.

scope. At test time, the rule with the highest score among the applicable rules is used.

4.1.4 Feature Invariant Transformer (FIT)

Wu et al. (2021) proposed a simple technique employing a character-level transformer for feature-guided transduction that was used as a baseline for the 2021 SIGMORPHON shared task.[5] This is a generative model capable of performing character-level decoding to generate target inflections. In comparison to a vanilla transformer model, positional counts are used only for characters and not for features. The model also incorporates unique tokens to mark whether a given token is a feature.

4.1.5 Principle Parts for Inflection (PPI)

We apply the approach of Liu and Hulden (2020), which recasts the task of morphological inflection as a "paradigm cell filling problem." This leverages a lexeme's principal parts—the minimum subset of paradigm slots needed to generate the other slots in its paradigm. Specifically, for low-resource scenarios, the principal parts of a paradigm identify additional slots that are crucial in generating the target-inflected lemma.

4.1.6 Analogical Encoder-Decoder (AED)

Following up on Albright and Hayes (2003) and Kirov and Cotterell (2018), Calderone et al. (2021) proposed a recurrent neural network encoder-decoder architecture augmented with pre-compiled analogical patterns for generating morphological inflections of nonce words. This model leverages the UniMorph Tags and fine alternation pattern (FAP) associated with each lemma in relation to its inflection form. FAPs analyze the positioning of word forms within the system to identify recurrent patterns representing conventional linguistic elements.

4.2 Prompting

We employ three distinct prompting styles, namely zero-, one-, and few-shot, to interact with the language model. We start with a simple instruction in each language, for example:

> "Fill in the blank with the correct past tense of the word 'wug'. Give your response in one word.
> They wug all the time. In fact, they ____ just yesterday."

For Tamil, the instruction portion of the prompt is omitted because of ChatGPT's unreliable performance when given instructions in that language. We select one example with real words for each major inflection class of the phenomenon in question. We then perform multiple runs: 10 for the zero-shot scenario, one for every shot for the one-shot scenario, and 10 for the few-shot scenario, with a new random permutation of all examples each time. We query `gpt-3.5-turbo-0613`, select the first word of the response, and filter by removing non-word characters. We evaluate by computing the accuracy for each of the runs, averaged over all queried nonce words, and compute the mean and standard deviation across all runs. We employ *acc@k* as our evaluation metric, setting $k = 5$ for our main evaluation. We provide results for $k = 1$ and $k = 3$ in Appendix A.4. The k gold forms are the k responses most frequently generated by humans. Since only one Turkish response is possible (the morphology is deterministic), k is always 1 for this language. We then perform an additional experiment for comparison in which we remove the context around the nonce word and only give the instructions as well as the last line. We call this the *short* prompt and the original described above the *long* prompt. We provide instances of *long* and *short* prompt in Appendix A.5.

[5] https://github.com/sigmorphon/2021Task0/tree/main/baselines

5 Results

5.1 Overall Performance

For *acc*@5, the performance of ChatGPT never exceeded that of the strongest baselines (ARL, AED, and PPI) regardless of the prompting regime, as shown in Table 5. However, it beats certain older baselines such as MinGen (the minimum generalization learner). ChatGPT performed best when it was explicitly prompted to complete an analogy with a single example (i.e., short 1-shot), as can be seen in Figure 2. We observe that similar trends hold for *acc*@1 and *acc*@3 (see Appendix A.4), but the gap between the strongest baselines and ChatGPT decreases with *k*.

English ChatGPT's performance on English was uniformly worse than both the average annotator (87.64%) and the strongest baselines. *acc*@1 falls below 60% in the 0-shot condition but is markedly better when shots are supplied. Short prompts, which require the model to complete a simple analogy, resulted in better performance than long prompts. In all conditions, authentic English words that did not occur in the reference annotations appeared as outputs when the nonce word and the authentic word were orthographically similar (see the discussion in Section 6.4).

German The best German result was 88.94% (short 1-shot), which beat all of the baselines except for ARL and FIT. The other results are similarly strong in contrast to the other languages. The impact of *k* is not noticeable here. This, in combination with the fact that the human performance on *acc*@5 was 88%, indicates that the task is perfectly performed by ChatGPT. It has reached the upper bound given by the inherent subjectivity of the task (reflected in the human variability) and the impact of *k* is, therefore, not measurable. This is further solidified by the very small impact of the long vs. short prompts.

Tamil Tamil performance of ChatGPT was significantly worse than the provided baselines, even in the few-shot conditions. For the few-shot case, there was marginally better performance when using short prompts, but this did not apply to the 0- or 1-shot case (in which no accurate outputs were generated). Across the board, the performance on Tamil was markedly worse than performance on English and German. However, considering that the average annotator had only 43.85% accuracy against the judgments of the other annotators, the few-shot accuracy is quite reasonable.

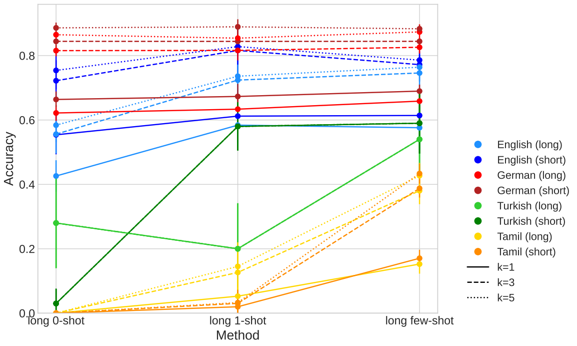


Figure 2: Results for the different prompt scenarios, formats, languages, and values of *k*.

Turkish The prompting performance for the Turkish inflection task is worse than for English and German, especially in the long prompt case. For this task, the morphotactics is trivial but the selection of the allomorph depends on stem vowels, stem-final consonants, whether there is a consonant cluster ending the stem, and whether the stem is monosyllabic or not. ChatGPT gets better results with the short prompt through an analogical example. For the three reinflection tasks, ChatGPT gets mixed results that are overall worse than for the inflection task (see Table 6).

6 Analysis

6.1 The Nature of the Task

The inherent complexity of the inflection tasks for the various languages (and the reinflection task for Turkish) varies greatly. English and Turkish are the simplest: the top-ranked form can always be obtained by adding a single suffix and applying a few morphographemic alternations. German annotations show no dominant pattern and assign nonce words to morphological classes according to complex criteria. However, German performance is clearly better, suggesting that factors other than inherent complexity play a role in ChatGPT's ability to generalize morphological patterns.

6.2 Impact of Tokenization

There is mounting evidence that the morphologically suboptimal nature of many tokenizers may limit the morphological capabilities of LLMs (Bostrom and Durrett, 2020; Hofmann et al., 2021). ChatGPT's tokenization, i.e., byte-pair encoding

Method	English	German	Tamil	Turkish
ARL	100.00	94.25	61.48	60.00
MinGen	62.00	64.37	49.18	40.00
FIT	98.00 ± 1.26	92.87 ± 0.74	63.28 ± 3.36	67.00 ± 4.58
PPI	94.60 ± 2.54	85.98 ± 5.91	55.33 ± 1.84	68.00 ± 4.00
AED	57.60 ± 6.62	48.51 ± 5.45	58.69 ± 5.46	56.00 ± 4.90
long 0-shot	58.40 ± 5.28	86.49 ± 1.07	0.00	28.00 ± 14.00
long 1-shot	73.60 ± 6.97	85.42 ± 2.52	14.52 ± 7.48	20.00 ± 14.14
long few-shot	76.40 ± 4.45	87.36 ± 2.37	42.70 ± 3.96	54.00 ± 10.20
short 0-shot	75.40 ± 5.87	88.62 ± 1.64	0.00	3.00 ± 4.58
short 1-shot	82.80 ± 5.60	88.94 ± 2.35	3.28 ± 3.99	58.00 ± 7.48
short few-shot	78.60 ± 2.84	88.33 ± 1.15	43.36 ± 3.12	59.00 ± 9.43

Table 5: Results (*acc@k*) for all languages ($k = 5$ except for Turkish where $k = 1$, cf. Section 4.2).

Type	0-shot	1-shot	few-shot
long	3.00 ± 1.80	20.67 ± 5.73	33.33 ± 4.94
short	7.00 ± 4.33	18.67 ± 6.18	31.00 ± 4.23

Table 6: Results for Turkish averaged over the three reinflection tasks ($k = 1$).

(Sennrich et al., 2016), has been shown to be particularly problematic (Bostrom and Durrett, 2020; Hofmann et al., 2022).

To examine the impact of tokenization, we measured the number of tokens into which the nonce words are split for the individual languages and computed the accuracy as a function of the number of tokens. Our hypothesis was that longer token sequences are less optimal, potentially leading to worse performance. However, using two-sided t-tests, we did not find a significant difference between nonce words with different token lengths. We interpret this as indicating that tokenization plays a less pronounced role for ChatGPT.

6.3 Impact of k

We observe that the gap between the baselines and our results increases with k (see Table 5, Appendix A.4), suggesting that ChatGPT tends to generate either a top-ranked form or an implausible inflection while the baselines tend to produce plausible inflections which are less frequent in the human annotations. ChatGPT's penchant for implausible inflections may be a result of its real word bias (see Section 6.4 below).

6.4 Real Word Bias

In English and German—and to a lesser extent in Turkish—many of the forms generated by ChatGPT belong to a different lexeme than the nonce word and thus do not constitute inflections in any strict linguistic sense (see Section 2.1). Crucially, the stem of the generated form is always a real word (i.e., a word that exists in the respective language). Examples of this phenomenon include, for English: *did* as the past tense of *dedo*, *blushed* as the past tense of *blus*, *fried* as the past tense of *fride*; and for German: *Ozeane* ('oceans') as the plural of *Ozeak*, *Institute* ('institutes') as the plural of *Instite*, *Sklaven* ('slaves') as the plural of *Schlave*. It is important to notice that in all these cases, (i) the generated form has the correct morphological properties—e.g., the English forms *did*, *blushed*, *fried* are indeed past tense forms—but the stem is a real word rather than the nonce word, and (ii) the stem that is generated in lieu of the nonce word is a frequently occurring word in the respective language and has a certain (sometimes strong) orthographic similarity to the nonce word. We denote this tendency *real word bias*.

The concept of real word bias allows us to make a hypothesis about the way in which ChatGPT addresses morphological tasks. We think ChatGPT is *not* applying morphological rules to a stem, which would be in line with item-and-process accounts of morphology (Hockett, 1954). Rather, it seems to linguistically decode the point in its representational space defined by the semantic constraints in the prompt. In cases where this point (and its immediate neighborhood) is unoccupied, it generates a form based on the nonce word, but in cases where there is a form of a real word close to the point (e.g., because of superficial orthographic similarity), it generates this form instead. The fact that the real word bias is strongest for German and English (the two high-resource languages) suggests that the representational space is more dense for these two languages, increasing the probability that there is a real word close to the point that the model is trying

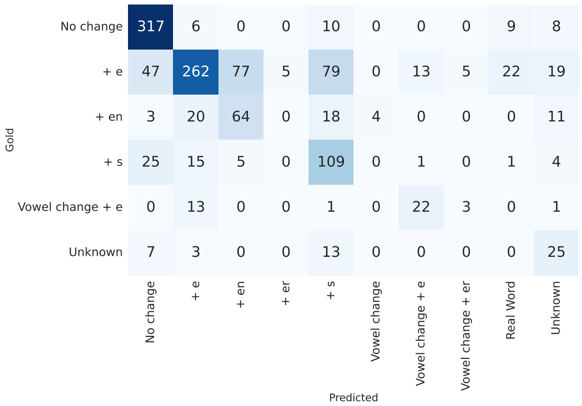


Figure 3: Confusion matrix for competing German plural morphemes for the few-shot setting.

to decode based on the prompt.

6.5 Morphological Productivity

The productivity of a morpheme is traditionally defined as its propensity to be used in novel combinations (Plag, 1999; Bauer, 2001; Haspelmath and Sims, 2010). Crucially, morphemes with the same meaning can differ in their productivity—for example, for English deadjectival nominalizing suffixes, *-ness* (e.g., *robustness*) is generally more productive than *-ity* (e.g, *equality*), which in turn is more productive than the fully non-productive *-th* (e.g., *warmth*). We are interested to see whether there is any difference in the productivity of morphological patterns exhibited by ChatGPT compared to the human sample. We focus on German as it has the most complex pattern of competing morphemes, and we examine the few-shot results as they show the best performance overall.

We start by comparing the distribution over alternative plural morphemes generated by ChatGPT with the human responses. As shown in Figure 3, there are several morphemes that are used by ChatGPT similarly to humans (e.g., the null morpheme). Cases of overgeneralization, where ChatGPT systematically generalizes the usage of a particular suffix to contexts where the suffix is not used by humans, are mainly limited to two plural morphemes: *-en* (77 generations for gold morpheme *-e*) and *-s* (79 generations for gold morpheme *-e*). Interestingly, these two plural morphemes are the two most productive plural morphemes in German (Köpcke, 1988). This indicates two important points: (i) ChatGPT is sensitive to the productivity of morphemes, i.e., it has acquired the ability to model how productive certain morphemes are as a result of pretraining; (ii) it does not identically mirror the behavior of humans, but rather amplifies the productivity of certain morphemes. The finding that the most productive morphemes (for humans) are becoming more productive for ChatGPT while the least productive morphemes (for humans) are becoming less productive for ChatGPT bears some theoretical resemblance to discussions about bias amplification (Ahn et al., 2022).

7 Future Directions

Morphological patterns are only one kind of generalization that can be investigated through a wug-like experimental paradigm. The form-meaning relationships encoded in language and multimodal models, including constructional and iconic pairings, can be investigated through prompting with nonce stimuli, leading to new insights regarding the generalizations they capture.

Limitations

Our research was conducted with a single model (`gpt-3.5-turbo-0613`), so it is not certain that our results will generalize to other versions of GPT-3 or to GPT-4, let alone other LLMs. Although we went to great lengths to develop prompts that would maximize ChatGPT's performance on the tasks, it is not possible to state definitively that another strategy would not produce better performance. While the languages were typologically varied, it is not clear whether the results observed in the current study are generally robust or are coincidental properties of the small set of languages and datasets under investigation. Furthermore, comparing the languages to one another is problematic because it was not possible to control other variables while varying the language. For example, the English and Tamil tasks involve verbal inflection while the German and Turkish tasks involve nominal inflection. Finally, the number of annotators for Tamil was very small and inter-annotator agreement was very low, meaning that the results of the Tamil experiments must be approached with special caution (but see our discussion about morphological variation in Section 3).

Ethics

LLMs are already impacting the world's people in significant ways, for good and ill. Understanding their limitations, particularly with regard to non-hegemonic language communities, is an ethical imperative. This study highlights one specific way in which an LLM should not be treated as a surrogate human, thus motivating additional research on language modeling for structurally diverse and low-resource languages.

Acknowledgements

This work was funded by the European Research Council (#740516). The second author was also supported by the German Academic Scholarship Foundation. We thank the reviewers for their extremely helpful comments.

References

S Agesthialingom. 1971. A note on Tamil verbs. *Anthropological Linguistics*, pages 121–125.

Jaimeen Ahn, Hwaran Lee, Jinhwa Kim, and Alice Oh. 2022. Why knowledge distillation amplifies gender bias and how to mitigate from the perspective of DistilBERT. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 266–272, Seattle, Washington. Association for Computational Linguistics.

Adam Albright and Bruce Hayes. 2002. Modeling english past tense intuitions with minimal generalization. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, MPL '02, page 58–69, USA. Association for Computational Linguistics.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, 90(2):119–161.

Albert Henry Arden. 1891. *A progressive grammar of common Tamil*. Society for Promoting Christian Knowledge.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, Jaime Rafael Montoya Samame, Delio Siticonatzi Camaiteri, Esaú Zumaeta Rojas, Didier López Francis, Arturo Oncevay, Juan López Bautista, Gema Celeste Silva Villegas, Lucas Torroba Hennigen, Adam Ek, David Guriel, Peter Dirix, Jean-Philippe Bernardy, Andrey Scherbakov, Aziyana Bayyr-ool, Antonios Anastasopoulos, Roberto Zariquiey, Karina Sheifer, Sofya Ganieva, Hilaria Cruz, Ritván Karahóǧa, Stella Markantonatou, George Pavlidis, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Candy Angulo, Jatayu Baxi, Andrew Krizhanovsky, Natalia Krizhanovskaya, Elizabeth Salesky, Clara Vania, Sardana Ivanova, Jennifer White, Rowan Hall Maudslay, Josef Valvoda, Ran Zmigrod, Paula Czarnowska, Irene Nikkarinen, Aelita Salchak, Brijesh Bhatt, Christopher Straughn, Zoey Liu, Jonathan North Washington, Yuval Pinter, Duygu Ataman, Marcin Wolinski, Totok Suhardijanto, Anna Yablonskaya, Niklas Stoehr, Hossep Dolatian, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Aryaman Arora, Richard J. Hatcher, Ritesh Kumar, Jeremiah Young, Daria Rodionova, Anastasia Yemelina, Taras Andrushko, Igor Marchenko, Polina Mashkovtseva, Alexandra Serova, Emily Prud'hommeaux, Maria Nepomniashchaya, Fausto Giunchiglia, Eleanor Chodroff, Mans Hulden, Miikka Silfverberg, Arya D. McCarthy, David Yarowsky, Ryan Cotterell, Reut Tsarfaty, and Ekaterina Vylomova. 2022. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Laurie Bauer. 2001. *Morphological productivity*. Cambridge University Press, Cambridge (UK).

Rachel Bawden, Jonathan Poinhos, Eleni Kogkitsidou, Philippe Gambette, Benoît Sagot, and Simon Gabay. 2022. Automatic normalisation of early Modern French. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3354–3366, Marseille, France. European Language Resources Association.

Jean Berko. 1958. The child's learning of English morphology. *Word*, 14(2-3):150–177.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Romina Brown, Santiago Paez, Gonzalo Herrera, Luis Chiruzzo, and Aiala Rosá. 2023. Experiments on automatic error detection and correction for uruguayan learners of English. In *Proceedings of the 12th Workshop on NLP for Computer Assisted Language Learning*, pages 45–52, Tórshavn, Faroe Islands. LiU Electronic Press.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4.

Basilio Calderone, Nabil Hathout, and Olivier Bonami. 2021. Not quite there yet: Combining analogical patterns and encoder-decoder networks for cognitively plausible inflection. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 274–282, Online. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. *Arxiv*, 2204.02311.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017a. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016a. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Ryan Cotterell, Arun Kumar, and Hinrich Schütze. 2016b. Morphological segmentation inside-out. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2325–2330, Austin, Texas. Association for Computational Linguistics.

Ryan Cotterell, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-Markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 164–174, Beijing, China. Association for Computational Linguistics.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016c. A joint model of orthography and morphological segmentation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 664–669, San Diego, California. Association for Computational Linguistics.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017b. Paradigm completion for derivational morphology. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 714–720, Copenhagen, Denmark. Association for Computational Linguistics.

Antje Dammel and Oliver Schallert. 2019. *Morphological variation: Theoretical and empirical perspectives*. John Benjamins, Amsterdam.

Daniel Deutsch, John Hewitt, and Dan Roth. 2018. A distributional and orthographic aggregation model for English derivational morphology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1938–1947, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Robert M. Dixon. 1994. *Ergativity*. Cambridge University Press, Cambridge, UK.

Daniel Edmiston. 2020. A systematic analysis of morphological content in BERT models for multiple languages. *Arxiv*, abs/2004.03032.

Alexander Erdmann, Micha Elsner, Shijie Wu, Ryan Cotterell, and Nizar Habash. 2020a. The paradigm discovery problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7778–7790, Online. Association for Computational Linguistics.

Alexander Erdmann, Tom Kenter, Markus Becker, and Christian Schallhart. 2020b. Frugal paradigm completion. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8248–8273, Online. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *Arxiv*, 1901.05287.

Omer Goldman, David Guriel, and Reut Tsarfaty. 2022. (Un)solving morphological inflection: Lemma overlap artificially inflates models' performance. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 864–870, Dublin, Ireland. Association for Computational Linguistics.

Coleman Haley. 2020. This is a BERT. now there are several of them. can they generalize to novel words? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 333–341, Online. Association for Computational Linguistics.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Martin Haspelmath and Andrea Sims. 2010. *Understanding Morphology*. Routledge, Oxford (UK).

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are GPT models at machine translation? a comprehensive evaluation. *Arxiv*, 2302.09210.

Charles F. Hockett. 1954. Two Models of Grammatical Description. WORD, 10(2-3):210–234.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020a. DagoBERT: Generating derivational morphology with a pretrained language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3848–3861, Online. Association for Computational Linguistics.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020b. Predicting the growth of morphological families from social and linguistic factors. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7273–7283, Online. Association for Computational Linguistics.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3594–3608, Online. Association for Computational Linguistics.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics.

Valentin Hofmann, Hinrich Schütze, and Janet Pierrehumbert. 2020c. A graph auto-encoder model of derivational morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1127–1138, Online. Association for Computational Linguistics.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. *arXiv preprint arXiv:2305.10160*.

Wenxiang Jiao, Wenxuan Wang, JT Huang, Xing Wang, and ZP Tu. 2023. Is ChatGPT a good translator? Yes with GPT-4 as the engine. *arXiv*, 2301.08745.

Huiming Jin, Liwei Cai, Yihui Peng, Chen Xia, Arya McCarthy, and Katharina Kann. 2020. Unsupervised morphological paradigm completion. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6696–6707, Online. Association for Computational Linguistics.

Zijia Jin, Xingyu Zhang, Mo Yu, and Lifu Huang. 2022. Probing script knowledge from pre-trained models. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 87–93, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 961–967, Austin, Texas. Association for Computational Linguistics.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. Neural multi-source morphological reinflection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 514–524, Valencia, Spain. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70, Berlin, Germany. Association for Computational Linguistics.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Klaus-Michael Köpcke. 1988. Schemas in German plural formation. *Lingua*, 74(4):303–335.

Leigh Lisker. 1951. Tamil verb classification. *Journal of the American Oriental Society*, 71(2):111–114.

Ling Liu and Mans Hulden. 2020. Leveraging principal parts for morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 153–161, Online. Association for Computational Linguistics.

Ling Liu and Lingshuang Jack Mao. 2016. Morphological reinflection with conditional random fields and unsupervised features. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 36–40, Berlin, Germany. Association for Computational Linguistics.

Xiaomeng Ma and Lingyu Gao. 2022. How do we get there? Evaluating transformer neural networks as cognitive models for English past tense inflection. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1101–1114, Online only. Association for Computational Linguistics.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Gary F Marcus, Ursula Brinkmann, Harald Clahsen, Richard Wiese, and Steven Pinker. 1995. German inflection: The exception that proves the rule. *Cognitive psychology*, 29(3):189–256.

Rowan Hall Maudslay and Ryan Cotterell. 2021. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 124–131, Online. Association for Computational Linguistics.

Kate McCurdy, Sharon Goldwater, and Adam Lopez. 2020. Inflecting when there's no majority: Limitations of encoder-decoder neural networks as cognitive models for German plurals. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1745–1756, Online. Association for Computational Linguistics.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148.

OpenAI. 2023a. ChatGPT. Large language model.

OpenAI. 2023b. GPT-4 Technical Report.

Tiago Pimentel, Maria Ryskina, Sabrina J. Mielke, Shijie Wu, Eleanor Chodroff, Brian Leonard, Garrett Nicolai, Yustinus Ghanggo Ate, Salam Khalifa, Nizar Habash, Charbel El-Khaissi, Omer Goldman, Michael Gasser, William Lane, Matt Coler, Arturo Oncevay, Jaime Rafael Montoya Samame, Gema Celeste Silva Villegas, Adam Ek, Jean-Philippe Bernardy, Andrey Shcherbakov, Aziyana Bayyr-ool, Karina Sheifer, Sofya Ganieva, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Andrew Krizhanovsky, Natalia Krizhanovsky, Clara Vania, Sardana Ivanova, Aelita Salchak, Christopher Straughn, Zoey Liu, Jonathan North Washington, Duygu Ataman, Witold Kieraś, Marcin Woliński, Totok Suhardijanto, Niklas Stoehr, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Richard J. Hatcher, Emily Prud'hommeaux, Ritesh Kumar, Mans Hulden, Botond Barta, Dorina Lakatos, Gábor Szolnok, Judit Ács, Mohit Raj, David Yarowsky, Ryan Cotterell, Ben Ambridge, and Ekaterina Vylomova. 2021. SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–259, Online. Association for Computational Linguistics.

Steven Pinker and Alan Prince. 1988. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28(1-2):73–193.

Ingo Plag. 1999. *Morphological productivity: Structural constraints in English derivation*. De Gruyter, Berlin.

Kim Plunkett and Patrick Juola. 1999. A connectionist model of English past tense and plural morphology. *Cognitive Science*, 23(4):463–490.

David E Rumelhart and James L McClelland. 1986a. On learning the past tenses of English verbs. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2, pages 216–271. MIT Press, Cambridge, MA.

David E. Rumelhart and James L. McClelland. 1986b. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. The MIT Press.

Harold F Schiffman and Vasu Renganathan. 2009. *An English dictionary of the Tamil verb*. Linguistic Data Consortium.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Vancouver. Association for Computational Linguistics.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, Denver, Colorado. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models.

Michael T Ullman, Suzanne Corkin, Marie Coppola, Gregory Hickok, John H Growdon, Walter J Koroshetz, and Steven Pinker. 1997. A neural dissociation within language: Evidence that the mental dictionary is part of declarative memory, and that grammatical rules are processed by the procedural system. *Journal of Cognitive Neuroscience*, 9(2):266–276.

Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin, and Trevor Cohn. 2017. Context-aware prediction of derivational word-forms. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 118–124, Valencia, Spain. Association for Computational Linguistics.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. *arXiv*, 2304.02210.

Leonie Weissweiler, Valentin Hofmann, Masoud Jalili Sabet, and Hinrich Schuetze. 2022. CaMEL: Case Marker Extraction without Labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5506–5516, Dublin, Ireland. Association for Computational Linguistics.

Colin Wilson and Jane S.Y. Li. 2021. Were we there already? Applying minimal generalization to the SIGMORPHON-UniMorph shared task on cognitively plausible morphological inflection. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 283–291, Online. Association for Computational Linguistics.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

Hongzhi Xu, Mitchell Marcus, Charles Yang, and Lyle Ungar. 2018. Unsupervised morphology learning with statistical paradigms. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 44–54, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

A Appendices

A.1 Morphological Tags

In Table 7, we provide details about the morphological tags that are comprised by the train, dev, test, and wug test sets for the four languages. The tags for English (eng), German (deu), and Tamil (tam) are defined in accordance with the description in UniMorph 4.0 dataset. For Turkish (tur),the tags are defined in Section 3.

A.2 Hyperparameter Tuning

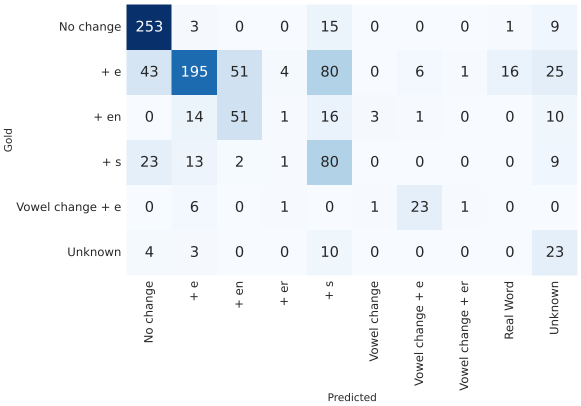
For all baselines, we follow the hyperparameter settings from the publicly available code repositories. The only exception is AED, where the number of epochs was increased from 40 to 200.

A.3 Qualtrics Details

Our study leveraged Qualtrics, a robust and comprehensive survey software tool that facilitates the

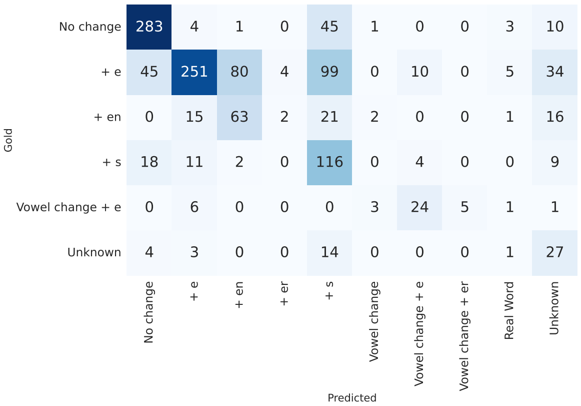
Lang	Tags	Train	Dev	Test	Wug
eng	V;NFIN	2015	206	202	0
eng	V;PRS;NOM(3,SG)	1987	190	213	0
eng	V;PST	1981	198	185	50
eng	V;V.PTCP;PRS	2018	201	200	0
eng	V;V.PTCP;PST	1999	205	200	0
deu	V.PTCP;PRS	246	19	19	0
deu	V;IMP;NOM(2,PL)	246	19	16	0
deu	V;IMP;NOM(2,SG)	241	22	17	0
deu	V;IND;PRS;NOM(1,PL)	246	20	18	0
deu	V;IND;PRS;NOM(1,SG)	227	21	17	0
deu	V;IND;PRS;NOM(2,PL)	250	29	21	0
deu	V;IND;PRS;NOM(2,SG)	258	25	15	0
deu	V;IND;PRS;NOM(3,SG)	233	21	22	0
deu	V;IND;PST;NOM(1,PL)	235	26	28	0
deu	V;IND;PST;NOM(1,SG)	236	17	11	0
deu	V;IND;PST;NOM(2,PL)	257	20	23	0
deu	V;IND;PST;NOM(3,PL)	243	15	29	0
deu	V;IND;PST;NOM(3,SG)	247	22	27	0
deu	V;NFIN	248	21	13	0
deu	V;SBJV;PRS;NOM(1,PL)	234	25	18	0
deu	V;SBJV;PST;NOM(1,PL)	243	20	20	0
deu	V;SBJV;PST;NOM(2,SG)	229	27	24	0
deu	V;SBJV;PST;NOM(3,PL)	247	22	22	0
deu	N;ACC(PL)	368	44	49	0
deu	N;ACC(SG)	385	47	53	0
deu	N;DAT(PL)	361	44	48	0
deu	N;DAT(SG)	382	47	52	0
deu	N;GEN(PL)	364	44	48	0
deu	N;GEN(SG)	385	47	50	0
deu	N;NOM(PL)	370	44	49	174
deu	N;NOM(SG)	391	47	53	0
deu	V.PTCP;PST	242	23	23	0
deu	V;IND;PRS;NOM(3,PL)	232	25	19	0
deu	V;IND;PST;NOM(2,SG)	215	25	18	0
deu	V;SBJV;PRS;NOM(2,PL)	248	20	23	0
deu	V;SBJV;PRS;NOM(3,PL)	247	26	26	0
deu	V;SBJV;PRS;NOM(3,SG)	238	26	24	0
deu	V;SBJV;PST;NOM(3,SG)	261	22	26	0
deu	V;SBJV;PRS;NOM(1,SG)	246	22	16	0
deu	V;SBJV;PST;NOM(1,SG)	238	21	28	0
deu	V;SBJV;PST;NOM(2,PL)	239	17	17	0
deu	V;SBJV;PRS;NOM(2,SG)	222	18	18	0
tur	V;POS;PAST;A1SG	2005	201	202	10
tur	V;NEG;NARR;A2PL	2005	201	202	10
tur	N;A3SG;P1SG;DAT	2170	214	214	10
tur	N;A3SG;PNON;ACC	2172	214	214	10
tam	V;PRS-1SG	67	16	0	0
tam	V;FUT-1SG	67	16	0	0
tam	V;PST-2SG	67	16	0	0
tam	V;PRS-2SG	67	16	0	0
tam	V;FUT-2SG	67	16	0	0
tam	V;PST-3SG.M	67	16	0	123
tam	V;PRS-3SG.M	67	16	0	0
tam	V;FUT-3SG.M	67	16	0	0
tam	V;PST-3SG.F	67	16	0	0
tam	V;PRS-3SG.F	67	16	0	0
tam	V;FUT-3SG.F	67	16	0	0
tam	V;PST-3SG.HON	67	16	0	0
tam	V;PRS-3SG.HON	67	16	0	0
tam	V;FUT-3SG.HON	67	16	0	0
tam	V;PST-1PL	67	16	0	0
tam	V;PRS-1PL	67	16	0	0
tam	V;FUT-1PL	67	16	0	0
tam	V;PST-2PL	67	16	0	0
tam	V;PRS-2PL	67	16	0	0
tam	V;FUT-2PL	67	16	0	0
tam	V;PST-3PL	67	16	0	0
tam	V;PRS-3PL	67	16	0	0
tam	V;FUT-3PL	67	16	0	0

Table 7: Distribution of tags over the different splits for the four languages.



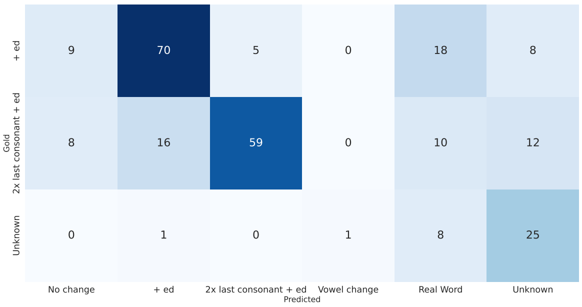
	No change	+ e	+ en	+ er	+ s	Vowel change	Vowel change + e	Vowel change + er	Real Word	Unknown
No change	253	3	0	0	15	0	0	0	1	9
+ e	43	195	51	4	80	0	6	1	16	25
+ en	0	14	51	1	16	3	1	0	0	10
+ s	23	13	2	1	80	0	0	0	0	9
Vowel change + e	0	6	0	1	0	1	23	1	0	0
Unknown	4	3	0	0	10	0	0	0	0	23

Figure 4: Confusion matrix for competing German plural morphemes for the one-shot setting.



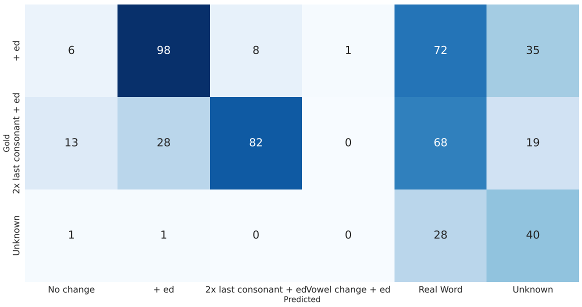
	No change	+ e	+ en	+ er	+ s	Vowel change	Vowel change + e	Vowel change + er	Real Word	Unknown
No change	283	4	1	0	45	1	0	0	3	10
+ e	45	251	80	4	99	0	10	0	5	34
+ en	0	15	63	2	21	2	0	0	1	16
+ s	18	11	2	0	116	0	4	0	0	9
Vowel change + e	0	6	0	0	0	3	24	5	1	1
Unknown	4	3	0	0	14	0	0	0	1	27

Figure 5: Confusion matrix for competing German plural morphemes for the zero-shot setting.



	No change	+ ed	2x last consonant + ed	Vowel change	Real Word	Unknown
+ ed	9	70	5	0	18	8
2x last consonant + ed	8	16	59	0	10	12
Unknown	0	1	0	1	8	25

Figure 6: Confusion matrix for competing English past tense morphemes for the one-shot setting.



	No change	+ ed	2x last consonant + ed	Vowel change + ed	Real Word	Unknown
+ ed	6	98	8	1	72	35
2x last consonant + ed	13	28	82	0	68	19
Unknown	1	1	0	0	28	40

Figure 7: Confusion matrix for competing English past tense morphemes for the zero-shot setting.

design of intricate online surveys.[6]

We initiated the survey by presenting an introduction that detailed the concept of a wug test and the associated information for the survey. This introductory passage served to inform participants of the nature and intent of the research study, and it also provided examples to further facilitate their understanding of our task requirements.

Our data collection phase consisted of two parts: the English wug test and the German wug test. Upon consenting to participate, respondents were guided through a series of thoughtfully designed prompts related to the wug test. These prompts encouraged them to provide suitable responses based on their understanding of the task.

For the English wug test, we employed the following exemplary prompt: “Fill in the blank with the correct past tense of the word ‘wug’. There is no predetermined correct answer. We encourage you to rely on your linguistic intuition. If you believe there are multiple possible responses, simply note the form that seems most accurate to you. For instance, ‘They wug all the time. In fact, they __ just yesterday!’”. Such prompts stimulated the participants to produce responses that were entirely their own, drawing on the provided information. For the German wug test, we translated the task instructions and prompts into German, ensuring easy comprehension for native German speakers.

In total, the English wug test incorporated 50 unique words for participants to respond to, while the German version consisted of 174 unique words. We received 28 responses for the English wug test and 21 responses for the German wug test.

A.4 Other Values of k

Table 8 presents results for $k = 1$ and $k = 3$. Results for $k = 5$ are given in Section 5.

A.5 Prompts

We leveraged the following prompts for the individual languages:

- English:
  - Long: “Fill in the blank with the correct past tense of the verb X. Answer with one word. They X all the time. In fact, they _ just yesterday! _ :”
  - Short: “Form the correct past tense of the verb X. Answer with one word. X :”
- German:
  - Long: “Fülle die Lücke mit dem korrekten Plural des Nomens X aus. Antworte mit einem Wort. Hier ist ein X. Jetzt sind es zwei _! _:”
  - Short: “Bilde den korrekten Plural des Nomens X. Antworte mit einem Wort. X :”
- Tamil:
  - Long: “நேற்று அவரிடம், "நீ X" என்றேன். அதைக் கேட்டு அவன் போய் _. _:”
  - Short: “X :”
- Turkish:
  - Long: “Boşlukları X ile verilen eylemin birinci tekil şahıs geçmiş zaman formları ile doldurun. Ben her zaman X. Ama dün _. _:”
  - Short: “Tek bir sözcük ile farazi X eyleminin birinci tekil şahıs geçmiş zaman hali nasıl olur? X :”

[6] https://www.qualtrics.com/

	English		German		Tamil	
Method	$k = 1$	$k = 3$	$k = 1$	$k = 3$	$k = 1$	$k = 3$
ARL	66.00	98.00	71.84	91.95	49.18	61.48
MinGen	56.00	60.00	39.66	60.92	39.34	49.18
FIT	84.00 ± 2.97	96.20 ± 0.60	70.06 ± 1.67	90.69 ± 0.99	44.75 ± 2.01	59.84 ± 3.07
PPI 1	72.60 ± 6.00	90.80 ± 4.49	60.17 ± 6.80	82.59 ± 6.56	37.30 ± 2.57	51.07 ± 1.56
AED	44.20 ± 7.18	56.20 ± 6.54	27.82 ± 3.94	42.87 ± 4.65	46.15 ± 4.18	57.87 ± 5.46
long 0-shot	42.60 ± 4.90	55.60 ± 5.99	62.18 ± 2.45	81.55 ± 1.77	0.00	0.00
long 1-shot	58.40 ± 7.20	72.40 ± 6.50	63.36 ± 4.01	81.61 ± 3.09	5.27 ± 2.83	12.65 ± 6.19
long few-shot	57.60 ± 6.97	74.60 ± 4.90	65.86 ± 3.03	82.59 ± 1.96	15.25 ± 2.98	38.03 ± 4.18
short 0-shot	55.40 ± 6.07	72.20 ± 6.35	66.38 ± 2.48	84.43 ± 2.63	0.00	0.00
short 1-shot	61.20 ± 8.16	81.60 ± 6.97	67.31 ± 3.92	84.41 ± 2.36	1.99 ± 2.47	3.04 ± 3.58
short few-shot	61.40 ± 3.69	77.20 ± 3.12	68.97 ± 2.02	84.43 ± 1.07	17.05 ± 2.64	38.77 ± 2.86

Table 8: Results for other values of *k*.

Chapter 11

Corresponds to the following publication:

Valentin Hofmann, **Leonie Weissweiler**, David R. Mortensen, Hinrich Schütze, Janet B. Pierrehumbert (2025): Derivational Morphology Reveals Analogical Generalization in Large Language Models. Proceedings of the National Academy of Sciences, U.S.A.

Declaration of Co-Authorship V.H., D.R.M., H.S., and J.B.P. designed research; V.H performed research; V.H. and L.W. collected data; V.H., L.W., D.R.M., H.S., and J.B.P. analysed data; J.B.P. synthesized prior literature; and V.H., L.W., D.R.M., and J.B.P. wrote the paper.

Research Context After the work presented in Chapter 10, the most natural extension is from morphological inflection to derivation. Derivation, in the following work specific from adjective to noun with either *-ity* or *-ness*, offers an ideal testbed to extend our field of research: we do not only ask how well the models have learned to generalise on this, we also investigate the underlying mechanism. This is possible for two reasons: 1) we have access to the weights of the language model, and its training data, allowing us to compare the frequencies of different suffixes to the production probabilities of the LM, and 2) the behaviour with regards to this derivation can be classified into several adjective classes, with varying degrees of homogeneity. These together allow for a more detailed assessment of the generalisation mechanisms in LLMs.

PNAS INAUGURAL ARTICLE | PSYCHOLOGICAL AND COGNITIVE SCIENCES
COMPUTER SCIENCES
OPEN ACCESS

Derivational morphology reveals analogical generalization in large language models

Valentin Hofmann[a,b,1], Leonie Weissweiler[c], David R. Mortensen[d], Hinrich Schütze[c], and Janet B. Pierrehumbert[e,1]

This contribution is part of the special series of Inaugural Articles by members of the National Academy of Sciences elected in 2019.
Contributed by Janet B. Pierrehumbert; received November 20, 2024; accepted February 18, 2025; reviewed by Joan Bresnan and Gary Dell

What mechanisms underlie linguistic generalization in large language models (LLMs)? This question has attracted considerable attention, with most studies analyzing the extent to which the language skills of LLMs resemble rules. As of yet, it is not known whether linguistic generalization in LLMs could equally well be explained as the result of analogy. A key shortcoming of prior research is its focus on regular linguistic phenomena, for which rule-based and analogical approaches make the same predictions. Here, we instead examine derivational morphology, specifically English adjective nominalization, which displays notable variability. We introduce a method for investigating linguistic generalization in LLMs: Focusing on GPT-J, we fit cognitive models that instantiate rule-based and analogical learning to the LLM training data and compare their predictions on a set of nonce adjectives with those of the LLM, allowing us to draw direct conclusions regarding underlying mechanisms. As expected, rule-based and analogical models explain the predictions of GPT-J equally well for adjectives with regular nominalization patterns. However, for adjectives with variable nominalization patterns, the analogical model provides a much better match. Furthermore, GPT-J's behavior is sensitive to the individual word frequencies, even for regular forms, a behavior that is consistent with an analogical account but not a rule-based one. These findings refute the hypothesis that GPT-J's linguistic generalization on adjective nominalization involves rules, suggesting analogy as the underlying mechanism. Overall, our study suggests that analogical processes play a bigger role in the linguistic generalization of LLMs than previously thought.

large language models | lexicon | analogy | linguistic rules | AI

In the recent past, large language models (LLMs) such as Chinchilla (1), Gemini (2), GPT-4 (3), LLaMA (4), Mistral (5), OLMo (6), and PaLM (7) have reached an unprecedented level of linguistic capability. While some have likened the language skills of LLMs to those of humans (8, 9), others have highlighted the persistent linguistic inadequacies of LLMs (10–13). Crucially, however, it is well established that they go beyond simply copying from the training data (14–19). With the ability to generate and process novel expressions being widely viewed as a hallmark of human intelligence, the controversy around the extent to which the language skills of LLMs are human-like has sparked a wider discussion about whether AI is finally truly intelligent.

What are the mechanisms underlying linguistic generalization in LLMs? Are they human-like? Prior studies have approached this question by investigating the extent to which the language skills of LLMs resemble abstract, symbolic linguistic rules (20, 21). For instance, the consistency with which LLMs provide the correct agreement marking for unseen subject–verb pairs has been interpreted as evidence that they implicitly infer a set of rules from the training data (18). Rules are a form of generalization that results from language learners scanning the available data and distilling abstract knowledge about the linguistic patterns exhibited in the data. Each rule has a structural description, which specifies what properties must be met for the rule to apply, and a structural change, which specifies how the input is changed to produce the output. When processing novel, previously unseen input, the rule matching the input properties is selected, and applied to produce the output.

Much less attention has been devoted to the question of whether the language skills of LLMs could be the result of analogical processes operating on stored exemplars. Typically expressed in the form $A : B :: C : D$ ("A is to B as C is to D"), an analogy is an assertion that the relation of A to B is similar to the relation of C to D. For example, the analogy *moon*:*planet*::*planet*:*sun* makes a generalization about our solar system, and entertaining this analogy was an important step in the Copernican development of the heliocentric theory. Analogical models of linguistic generalization take the D position

Significance

Large language models (LLMs) are a type of artificial intelligence technology that is currently being deployed in a rapidly growing range of applications. The sensitive nature of some of these applications makes it imperative that we have a precise understanding of the inner workings of LLMs. By uncovering the role of analogical mechanisms for the linguistic generalization of LLMs, our study contributes to this goal and casts a light on their impressive language skills. Furthermore, the results of our experiments have the potential to indicate pathways for further improving LLMs.

Author affiliations: [a]Allen Institute for AI, Seattle, WA 98103; [b]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA 98195; [c]Center for Information and Language Processing, Ludwig Maximilian University of Munich, Munich 80538, Germany; [d]School of Computer Science, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213; and [e]Oxford e-Research Centre, Department of Engineering Science, University of Oxford, Oxford, OX1 3QG, United Kingdom

Author contributions: V.H., D.R.M., H.S., and J.B.P. designed research; V.H performed research; V.H. and L.W. collected data; V.H., L.W., D.R.M., H.S., and J.B.P. analyzed data; J.B.P. synthesized prior literature; and V.H., L.W., D.R.M., and J.B.P. wrote the paper.

Reviewers: J.B., Stanford University; and G.D., University of Illinois.

The authors declare no competing interest.

[1]To whom correspondence may be addressed. Email: valentinh@allenai.org or janet.pierrehumbert@oerc.ox.ac.uk.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2423232122/-/DCSupplemental.

Published May 9, 2025.

in the template to be an unknown and add a level of statistical inference to find the optimal way to fill the *D* position. Specifically, for a probe *C* with unknown *D*, they ask which of the potential completed analogies for *C* has the best statistical support, as defined by the behavior of exemplars in memory that are similar to *C* (referred to as the neighborhood of *C*). Analogical models of word formation have proved successful in capturing detailed patterns of variation in multiple languages (22–28). For example, an analogical model can explain why the past tense of an English nonce verb *spling* is often judged to be *splang* (even though an *-ed* past tense is more common), based on the behavior of verbs in its neighborhood such as *sing*, *ring*, and *sink* (27).

Within cognitive science, analogical generalization is argued to be a central learning mechanism and a foundation for the ability of humans to form abstract conjectures (29–32). In evaluating the LLMs' analogical capabilities, we thus share with prior work the goal of evaluating the LLMs' capabilities for implicit abstraction. However, there are significant differences between analogical and rule-based theories of human reasoning, particularly with regard to effects of frequency. By generalizing on the fly over stored exemplars, analogical models have both the forest and the trees, in the form of both generalizations (forests) and trees (individual exemplars). As a result, frequency effects at multiple levels from individual known words to the overall prevalence of different patterns are predicted. Frequency effects do appear in rule-based theories in that the learner must encounter a sufficient mass of examples to learn a rule in the first place. However, the mental lexicon is only a repository of unpredictable information, which means that once a regular rule has been learned, its outputs are not stored. For example, the lexicon would include *rotate*, since the association of this word form with this specific concept could not have been predicted. However, it does not include the regular past tense *rotated*. The frequency of *rotated* would not be available in the model, neither as a factor in forming the past form of *rotate*, nor as an influence on the output for other base forms. This central difference between analogical models and rule-based models means that frequency effects are a repeated theme in the sections below.

LLMs store a considerable amount of their training data in their model weights (19, 33–35), thus implicitly providing a reservoir of stored exemplars that might support analogical reasoning as a mechanism for all generalizations. However, it is also possible that the models memorize examples but generalize only via rules. A third possibility is that LLMs learn rules for regular linguistic phenomena, while handling irregular linguistic phenomena by means of analogy over stored exemplars, in line with dual-mechanism approaches (27, 36–38). Thus, the way that the stored data are used in generalizing by LLMs is an open question.

Here, we present in-depth analysis of the role of analogical linguistic generalization in LLMs. Our work is motivated by a key shortcoming of the existing literature: Prior studies, most of which explore rule-based generalization in LLMs (e.g., ref. 18), have focused on syntactic phenomena such as subject–verb agreement, which display a high degree of regularity. Crucially, in such cases, both rule-based and analogical, exemplar-based approaches make the exactly same predictions (39–41); in other words, rule-like behavior of LLMs on regular linguistic phenomena does not represent any evidence for rule-based generalization. This very insight was at the heart of the pioneering research that first applied neural networks in the context of language learning, which argued that "lawful behavior and judgments may be produced by a mechanism in which there is no explicit representation of the rule" (42). In fact, neural network models of language depend in important ways on similarity relations among input examples (43–47), suggesting that analogy might play a major role for the language skills of LLMs. This hypothesis has not been systematically tested previously.

We focus on a domain of language that is known to exhibit more variability than syntax, making it better suited for distinguishing rule-based from analogical generalization: derivational morphology (48–52). Specifically, we analyze how LLMs learn English adjective nominalization with *-ity* and *-ness* (53–56), focusing on adjectives that themselves contain a derivational suffix (e.g., *available*, *selfish*, *hyperactive*). Such cases of affix stacking are an ideal testbed for our purposes since the adjective class (i.e., the adjective-final suffix) provides a controlled way to vary the regularity of the nominalization process: While some adjective classes are nominalized in a very regular way, exhibiting a clear preference for either *-ity* (e.g., adjectives ending in *-able* such as *available*) or *-ness* (e.g., adjectives ending in *-ish* such as *selfish*), others exhibit a substantial degree of variability (e.g., adjectives ending in *-ive* such as *hyperactive*). Furthermore, English adjective nominalization with *-ity* and *-ness* has been shown to be fully explainable as a result of analogical generalization in humans (57), suggesting that LLMs might employ the same mechanism. In general, probabilistic models (58), and particularly exemplar-based analogy models (59, 60), have recently proven very successful at modeling competition between nearly synonymous linguistic structures, which is an additional motivation for our work. While there has been some previous work on the morphological capabilities of LLMs (e.g., refs. 13, 15, and 61–63), it has not diagnosed the generalization mechanisms underlying those capabilities.

As a key contribution of our work, we introduce a method for probing the generalization mechanisms underlying the language skills of LLMs: We fit cognitive models that instantiate certain generalization mechanisms to the LLM training data and compare their predictions on unseen data with those of the LLM. This approach, which is inspired by a long line of research in computational psychology using computer simulations (64–66), adds to the growing body of work that seeks to explain the behavior of LLMs as a result of the data on which they were trained (18, 67–70). Our method also informs the LLMs that we analyze. Our primary target is GPT-J (71). GPT-J is one of the GPT series of generative pretrained transformer models, and it is one of the few LLMs whose training data, namely the Pile (72), is publicly available. We also present some results on GPT-4, as an example of a state-of-the-art model, even though lack of access to its training data creates some limitations. For the sake of convenience, we provide short definitions of the technical terms used throughout the paper in Table 1.

Results

Generalization to Nonce Words. We compare the linguistic generalization behavior of GPT-J with that of two high-performing cognitive models: the Minimal Generalization Learner (MGL; 73, 74) and the Generalized Context Model (GCM; 29, 30, 75). The MGL is a rule-based model that we have selected because it undertakes to capture detailed patterns of variation that earlier rule-based models did not capture, by assigning statistical reliability to rules. The GCM is an exemplar-based analogy model that was developed for perceptual categorization, and then successfully adapted to variability in inflectional morphology (24, 27). The two models are similar in that both generalize over word pairs consisting of a base form and a derived form,

Table 1. Key technical terms used in the paper, as they apply to the domain of word-formation

Adjective class	Set of adjectives ending in the same suffix.
Adjective nominalization	Derivational morphology that converts adjectives to nouns.
Analogy	Inference of a new word form *D* from word forms *A*, *B*, and *C* such that *C* is phonologically similar to *A*, and *D* is to *C* as *B* is to *A*.
Derivational morphology	Operations that change the meaning or part of speech of a word.
Derivative	Word form obtained by applying derivational morphology.
Exemplar	A specific instance of an item that is stored in memory.
Frequency	Count of a word or set of words in the corpus.
Neighborhood	Exemplars with a high phonological similarity to a probe.
Nonce word	Pseudoword invented for the purposes of an experiment.
Probability	Likelihood of a form as computed by a model.
Probe	Word form for which a derived word form is to be generated.
Rule	Statement of a pattern in which the output depends only on a symbolic description of the input.

and generate predictions for a novel derived form by mapping the phonological form of the base to the phonological form of the derivative. They can be trained on either word types or word tokens. The inventory of word types corresponds to the list of words in a mental lexicon; only the existence of a word in the language is taken into account, and not its frequency in the training data. In an inventory of word tokens, each occurrence of a word in the training data is treated separately, with the result that more frequent words have more instances than less frequent words. We consider both settings since the contrast between behaviors governed by type frequencies and those governed by token frequencies is a major theme in cognitive research on the lexicon (e.g., ref. 74).

The mechanics of GPT-J is substantially different. GPT-J has been trained on a large corpus of text, encoded as sequences of words and subunits of words (e.g., individual characters). The input and output of GPT-J both consist of text that can span several hundred words. To probe the implicit world knowledge of a model such as GPT-J, we can ask it to generate text answering questions about real-world facts. Similarly, to probe the model's implicit knowledge of derivational morphology, we can ask it to answer questions about the derived forms corresponding to a variety of base forms.

We focus on English adjective nominalization and examine four adjective classes (i.e., sets of adjectives ending in the same suffix), two of which clearly prefer *-ity* or *-ness* (specifically, adjectives ending in *-able* or *-ish*), and two of which are less regular while still showing an overall tendency toward one of the two suffixes (specifically, adjectives ending in *-ive* or *-ous*). We train the cognitive models on all adjective–derivative pairs that meet the following three criteria: i) the adjective belongs to one of the four adjective classes in question; ii) the derivative ends in *-ity* or *-ness*; iii) both the adjective and the derivative occur in the Pile.

For evaluation, we use UniPseudo (76) to generate 50 nonce adjectives for each of the four adjective classes. We check that both the generated nonce adjective and the two corresponding derivatives have a frequency of zero in the Pile, i.e., they have never been seen by either the cognitive models or GPT-J, thus providing an ideal test set for probing linguistic generalization. We then feed all nonce adjectives into the cognitive models and determine which of the two competing derivatives they prefer. For GPT-J, we measure the probability that it assigns to the two derivatives resulting from adding *-ity* and *-ness* to the adjectives. Specifically, we use GPT-J's autoregressive language modeling head to compute the log probabilities for the subword units into which the derivatives are split and sum them. We take the derivative with the higher total log probability as the preferred one. Since prior research has shown that varying prompts (i.e., the texts used to elicit LLM responses) can heavily affect LLM behavior (77), we repeat this procedure with 12 different prompts (*SI Appendix*, Supporting Text). If not stated otherwise, the presented results are averaged over prompts.

As shown in Fig. 1, both MGL and GCM—in the type-based as well as the token-based setting—make completely consistent predictions for the two adjective classes that strongly prefer one affix. They always predict *-ity* for *-able* and *-ness* for *-ish*. Thus, both cognitive models reproduce the regular behavior that characterizes these two adjective classes. GPT-J also predicts *-ity* for *-able*, and it predicts *-ness* for *-ish* in all but two cases for just one of the prompts (*turgeishity* and *prienishity*). GPT-J is nearly as successful in capturing the regular cases as MGL and GCM, and these in turn match the predictions of GPT-J equally well (Table 2, *Upper* panel). Thus, the regular adjective classes do not tell us whether GPT-J is more like a rule-base model or an analogical model.

Moving to the two adjective classes that show more variability between *-ity* and *-ness* (i.e., *-ive* and *-ous*), both MGL and GCM generate variable outcomes with a higher rate of *-ness* for *-ous* than for *-ity* (Fig. 1). However, the predictions differ substantially in detail: the cognitive models (in the type-based as well as the token-based setting) agree in only 54% of the adjective types. Crucially, the cognitive model that matches the predictions of GPT-J on these two adjectives classes best is the token-based GCM model (Table 2, *Lower* panel). As a concrete example, we consider the nonce adjective *pepulative*. The MGL models map *pepulative* to a rule that prescribes *-ity* following *-tive*, which in the type-based as well as the token-based setting has the highest confidence of all competing rules and is hence selected by both MGL models. The GCM models, by contrast, are more strongly influenced by local similarity effects. While overall there are a larger number of *-ity* derivatives in the neighborhood of *pepulative* (e.g., for adjectives ending in *-lative* there are 88 derivatives with *-ity* vs. 27 with *-ness*), many of the adjectives particularly close to *pepulative* have *-ness* derivatives with a high token frequency (e.g., *manipulativeness* has a token frequency of 1,544 vs. 26 for *manipulativity*). This difference is reflected by the GCM models, where the type-based model predicts *-ity*, but the token-based model predicts *-ness*. GPT-J, on the other hand, prefers *-ness* for this example and hence matches the behavior of the token-based GCM model.

Our results show that the generalization behavior of LLMs on linguistic phenomena with a high degree of variability is best explained as a result of analogical mechanisms. This finding is in line with the observation that LLMs store a considerable amount of their training data in their model weights (19, 33–35), and

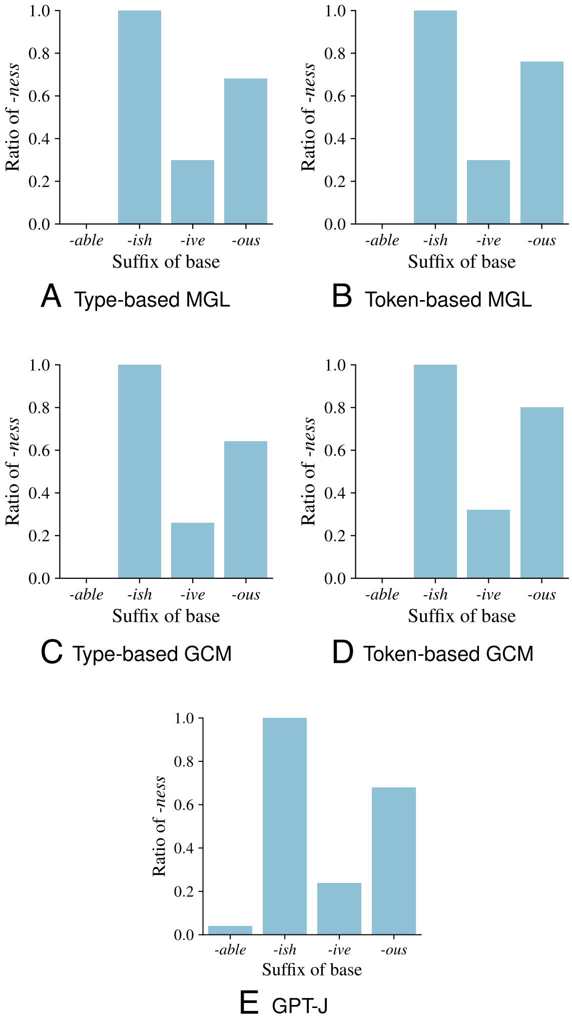


Fig. 1. Distribution of preferred nominalization type (specifically, ratio of *-ness* derivatives) for unseen nonce adjectives, for rule-based models (*A* and *B*), exemplar-based models (*C* and *D*), and GPT-J (*E*). Models based on types are shown on the *Left* (*A* and *C*), and models based on tokens are shown on the *Right* (*B* and *D*). The ratio is computed as the number of *-ness* predictions divided by the total number of predictions (i.e., *-ness* and *-ity* predictions).

it further suggests that these stored data actively contribute to the language skills displayed by LLMs. Our results are consistent with a model that generalizes all adjective nominalizations by analogy; they eliminate the possibility that all nominalizations are generated by rules. However, there remains the possibility that LLMs effectively use analogies in cases of variation and apply rules for adjective classes with a high degree of regularity. This possibility is suggested by earlier theories of inflectional morphology, proposing dual-mechanism models in which regular plurals and past tenses are created by rules, while irregular forms involve analogies (36–38). To address this possibility, it is necessary to look into frequency effects for individual words, as discussed in prior work (78, 79). We will do so in the next sections.

Predictions for Seen Words. According to cognitive theories, analogies are based on remembered examples. If the mechanism underlying GPT-J's behavior is analogical, it must implicitly remember a large number of examples. As the first step in evaluating this inference, we ask how well GPT-J's behavior matches the frequencies of words seen in its training data. Accurately matching the training data, derivative by derivative, would imply that the distributed representations in GPT-J encode information about individual derivatives.

We extend the four adjective classes examined so far and include six other adjective classes that can be nominalized with either *-ity* or *-ness*: *-al*, *-ar*, *-ed*, *-ic*, *-ing*, and *-less*. We can divide the ten adjective classes into four groups with similar degrees of competition between *-ity* and *-ness* (*SI Appendix*, Table S2):

- *-ed*, *-ing*, *-ish*, *-less* (R-NESS): This group exhibits the highest degree of regularity and almost always takes *-ness*.
- *-able*, *-al*, *-ar*, *-ic* (R-ITY): This group also exhibits a high degree of regularity (although somewhat lower than in the case of R-NESS), with a strong tendency toward *-ity*.
- *-ous* (V-NESS): This adjective class exhibits a high degree of variability, with a slight tendency toward *-ness*.
- *-ive* (V-ITY): This adjective class also exhibits a high degree of variability, with a slight tendency toward *-ity*.

We ask whether GPT-J treats adjectives from these four groups differently, and whether differences between the more regular and more variable ones correspond to differences in the training data. We draw upon the Pile and extract all derivatives ending in *-ity* and *-ness* whose bases belong to one of the 10 adjective classes. To decrease noise, we only extract derivatives whose bases also occur in the Pile and apply several filtering heuristics, such as excluding words with nonalphabetic characters. To include all productively formed derivatives, we do not impose a frequency threshold on the derivatives.

The overall setup of probing GPT-J is identical to the comparison with the cognitive models: We measure the probability that GPT-J assigns to the two derivatives resulting from adding *-ity* and *-ness* to the adjectives, using the same set of prompts. Following this procedure, we evaluate GPT-J on all 48,995 bases

Table 2. Comparison with cognitive models

		Examples		Counts		MGL		GCM	
Regularity	Suffix	Real	Nonce	*-ity*	*-ness*	Type	Token	Type	Token
High	*-able*	available	tegornable	11,081	1,034	0.893	0.893	0.893	0.893
	-ish	selfish	friquish	0	1,502	0.997	0.997	0.997	0.997
Low	*-ive*	sensitive	cormasive	4,508	2,438	†0.658	0.662	†0.622	**0.688**
	-ous	luminous	momogorous	1,372	2,450	†0.657	†0.613	†0.610	**0.703**

The table shows real and nonce examples for the four examined adjective classes, the counts of corresponding derivatives in the Pile as well as the results of rule-based and exemplar-based analogy models evaluated against GPT-J. Specifically, the choice of *-ity* or *-ness* for each nonce word by each of the four models shown is compared to GPT-J's choice for that item. The evaluation measure is accuracy, i.e., the percentage of the model's choices that matched GPT-J's choice. We highlight the highest accuracy value (i.e., the best-matching cognitive model) in each row in boldface—for the two adjective classes where there is a winner (i.e., *-ive* and *-ous*), this is the token-based GCM model. We highlight accuracy values that are significantly ($P < 0.05$) worse than the highest accuracy value in each row with a †.

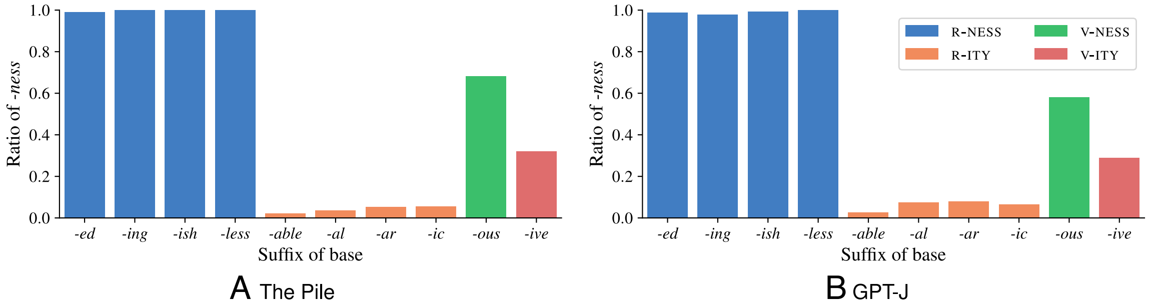


Fig. 2. Ratio of bases preferring *-ness* in the Pile (*A*) and GPT-J's predictions with one example prompt (*B*). Results are similar for the other prompts. The suffixes of the base (i.e., adjective classes) are grouped by degree of competition between *-ity* and *-ness*.

from the Pile. If not stated otherwise, results are again averaged across prompts.

Fig. 2 compares, for each adjective class, the ratio of bases for which GPT-J prefers *-ness* compared to *-ity* with the statistics from the Pile. We find that the two distributions are very similar: almost no competition for the bases in R-NESS (i.e., *-ed*, *-ing*, *-ish*, *-less*), little competition for the bases in R-ITY (i.e., *-able*, *-al*, *-ar*, *-ic*), and strong competition for V-NESS (i.e., *-ous*) and V-ITY (i.e., *-ive*). The tendency toward *-ity* and *-ness* is also exactly as predicted based on the training data—the average correlation between the class-level *-ity*/*-ness* ratios in the training data (Fig. 2*A*) and GPT-J predictions (Fig. 2*B*) is 0.995 (±0.004; $P < 0.001$ for all prompts), measured using Pearson's r. For multiple comparisons, P-values are corrected using the Holm–Bonferroni method (80).

Furthermore, GPT-J matches the training data statistics even on the level of individual bases: Across all bases, the accuracy of GPT-J's preference for one of the two derivatives compared against the training data (considered here as the ground truth) is 89.5% (±4.8%); the derivative preferred by GPT-J is generally the derivative that is more likely in the training data.

Table 3 shows that there is variation between individual adjective classes, with bases in R-NESS (*-ed*, *-ing*, *-ish*, *-less*) having above 95% accuracy, bases in R-ITY (*-able*, *-al*, *-ar*, *-ic*) having above 85% accuracy, and bases in V-ITY (*-ive*) and V-NESS (*-ous*) having below 85% accuracy, but the general level of agreement is very high.

Thus, GPT-J's morphological preferences closely mirror the statistics of the data it was trained on. The fact that GPT-J very consistently prefers the derivative with the higher frequency in the training data, even in cases such as adjectives ending in *-ive* where the suffix alone is a bad predictor of *-ity* vs. *-ness*, suggests that it stores many derivatives in its model weights. This is again in line with an analogical mechanism.

However, it is still possible that some of the high-regularity adjective classes (e.g., *-ish*) are handled by a rule, as suggested by dual-mechanism approaches. Next, we will disentangle these two hypotheses.

Frequency Effects and Neighborhood Effects. To further test whether at least part of GPT-J's behavior on adjective nominalization can be explained by rules, we analyze the extent to which GPT-J prefers an observed nominalized form over an alternative, nonobserved nominalized form. We consider only cases in which just one outcome of nominalization is attested in the Pile and measure the difference in the log probability that GPT-J assigns to the attested vs. the unattested form. This difference can be viewed as reflecting GPT-J's confidence in using a form that it has encountered during training; a large difference indicates high confidence, while a small difference reflects low confidence. For each adjective class, we create two sets: one in which the attested derivative has a low frequency in the Pile, $f \in (0, 10]$, and one in which the attested derivative has a high frequency in the Pile, $f \in (100, \infty)$.

If an adjective class is handled by a rule, the difference in frequency between the two sets should not affect GPT-J's confidence in predicting the attested derivative. This is because rule-based theories abstract away from individual words; once a rule has been acquired, regular complex forms are assumed to be generated on the fly, much like complex sentence structures are, rather than being stored in memory. Does this match GPT-J' behavior for any of the adjective classes? We operationalize this question by i) measuring GPT-J's confidence (i.e., the log probability difference between the attested and the unattested derivative) for low-frequency derivatives with $f \in (0, 10]$, and ii) measuring the relative increase in confidence for high-frequency derivatives with $f \in (100, \infty)$. If an adjective class is handled by a rule, this relative increase should be zero. We again divide the adjective classes into the four regularity-based groups defined above (R-NESS, R-ITY, V-NESS, and V-ITY).

Fig. 3 displays the results. The relative increase in confidence is positive for all adjective classes and for all prompts, indicating that GPT-J is always more confident in its decision for the frequent than the rare derivatives, even for the R-NESS class. This indicates that the model has stored distributed representations for all the derivatives, contrary to the predictions of dual-mechanism models. Put differently, none of the adjective classes are handled by rule.

Overall, the examined adjective classes exhibit a downward slope in Fig. 3, which is also reflected by the R-NESS and R-ITY groups individually (indicated by trendlines). Thus, the more

Table 3. Match between preferred derivatives in the training data and derivatives preferred by GPT-J

Adjective class	Suffix	Accuracy
R-NESS	*-ed*	0.986 ± 0.007
	-ing	0.989 ± 0.014
	-ish	0.995 ± 0.004
	-less	0.999 ± 0.001
R-ITY	*-able*	0.896 ± 0.082
	-al	0.884 ± 0.073
	-ar	0.896 ± 0.060
	-ic	0.867 ± 0.090
V-NESS	*-ous*	0.788 ± 0.038
V-ITY	*-ive*	0.842 ± 0.012

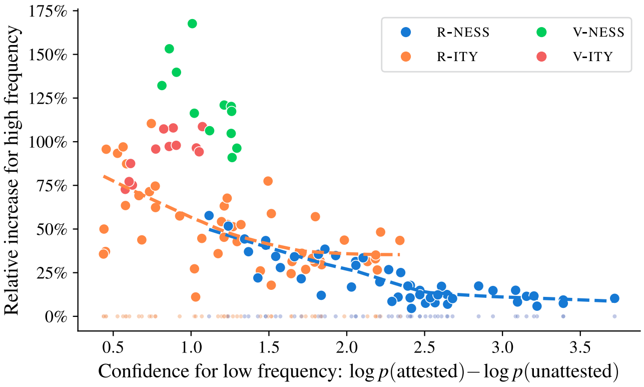


Fig. 3. Impact of word frequency on GPT-J's confidence in its choice. x-axis: Log probability difference between the attested and unattested choices for low-frequency derivatives with $f \in (0, 10]$. We have converted the log probabilities from base e to base 10 for better readability. y-axis: Relative increase in confidence for high-frequency derivatives with $f \in (100, \infty)$. Each dot corresponds to GPT-J's predictions for an adjective class given a specific prompt. Dots are colored by degree of competition between *-ity* and *-ness*. We added LOWESS lines for R-NESS and R-ITY. Dots at $y = 0\%$ indicate the expected behavior if R-NESS and R-ITY were handled by rule.

confident the model was in its decisions for the low-frequency group, the smaller the effect of word frequency on confidence. This finding is difficult to explain in a rule-based model, but it is perfectly in line with analogy as the underlying generalization mechanism.

More specifically, we will attribute the downward slope to the fact that the neighborhoods for probes from different adjective classes show varying degrees of competition between *-ity* and *-ness*. For the most regular adjective classes on the right-hand side of Fig. 3, there is little competition between *-ity* and *-ness* in the neighborhood for any given probe; for example, for an adjective ending in -ish, all adjectives in its neighborhood are nominalized with *-ness* (cf. Table 2), and hence the model confidence is high even if the attested derivative has a low frequency. In other words, if a derivative is strongly encoded in the model weights due to high frequency in the training data, this does not increase model confidence, because a highly homogeneous neighborhood already provides a clear signal as to which form should be preferred.

Lower confidence levels for the low-frequency forms toward the left of Fig. 3 represent cases in which the neighborhood of the probe is more heterogeneous. Here, the neighborhood alone provides a less clear signal as to which of the two alternative derivatives should be preferred; for example, for an adjective ending in *-ive*, its neighborhood often contains both adjectives nominalized with *-ness* and adjectives nominalized with *-ity* (cf. Table 2). As a result, the frequency in the training data becomes a critical factor for model confidence. If the frequency of the attested derivative is low, it is only weakly encoded in the model weights. Consequently, the model must rely on a heterogeneous neighborhood, which results in low confidence. On the other hand, if the frequency of the attested derivative is high, the resulting encoding in the model weights is strong, thus providing a clear signal beyond the neighborhood and leading to high confidence.

To quantify the neighborhood effect, we calculate the Shannon entropy of the distribution over *-ity* and *-ness* as the preferred form in the Pile for each adjective class. This serves as a rough approximation of the competition between *-ity* and *-ness* that is expected to exist in the neighborhoods of probes from each adjective class. We then use Pearson's r to measure the correlation between the entropy and the confidence increase for high-frequency derivatives. At $r^2 = 0.75$, $P < 0.001$, this correlation is highly significant. Thus, the more heterogeneous the neighborhoods of probes from an adjective class, the greater the impact of the attested derivative's frequency on GPT-J's confidence—a finding that is exactly in line with the predictions of analogical models (e.g., ref. 81) while being completely at odds with rule-based approaches, which do not assume such frequency effects to begin with.

The left side of Fig. 3 exhibits more variability than the right side. We believe that this variability is caused by local neighborhood effects and the interaction of these effects with the prompting mechanism. Recall from the discussion of the nonce word *pepulative* that analogical models are sensitive not merely to the overall statistics for the two competing nominalizations, but also to the similarity and frequency of the most similar neighbors. These localized effects—which for the case of attested derivatives would also include semantic similarity—create a lumpy prediction landscape whose properties we do not try to quantify here. Meanwhile, the prompting mechanism is known to influence the focus and bias of the underlying transformer model (82). Slightly different prompts direct the focus toward different parts of the lumpy landscape, and would hence produce noise in the datapoints for Fig. 3.

To sum up, our analysis suggests that GPT-J learns adjective nominalization by implicitly storing derivatives in its model weights. In cases where the exemplar neighborhood for a probe is highly homogeneous, GPT-J produces highly regular outputs. While regular, or rule-like, behavior of LLMs has been observed before (e.g., ref. 18), our results contextualize this finding in important ways, suggesting that rule-like behavior forms the end of a gradient characterized by varying levels of regularity. This result is not consistent with assuming a qualitative difference between forms derived by rule and stored exemplars. However, it is exactly in line with the predictions of exemplar-based analogy models (e.g., refs. 24, 27, and 28).

Human Use of Word Types vs. Tokens. We have established that GPT-J relies on token-level analogical generalization. In contrast, previous studies have concluded that humans generalize over word types (83–85): Their propensity to generalize a word formation pattern depends on the number of distinct word types in the individual's mental lexicon that support the pattern (referred to as the size of the lexical gang). This points to a difference between the morphological processing in humans and LLMs. We will now investigate this difference in greater detail, by comparing the predictions of GPT-J to judgments made by humans.

Judgments of nonce words. First, we make a direct comparison to GPT-J's behavior for nonce words. 22 native English speaker volunteer annotators indicated their preference for the *-ity* vs. the *-ness* derivative of each nonce adjective in our study. Because GPT-J is not a state-of-the-art model, we also introduce an additional comparison, by asking whether a more recent model is more human-like in its judgments. Specifically, we evaluate GPT-4 (3) on the same set of adjectives. If GPT-4 displays more human-like judgments than GPT-J, then the trend of improving LLMs through larger training sets and bigger model sizes will have paid off in this domain.

In Table 4, we take the derivative more often selected by humans as the ground truth. The table gives the accuracy of GPT-J, GPT-4, as well as the cognitive models considered above (i.e., MGL and GCM), measured against this human response. The type-based GCM model overall matches the human behavior best. While all cognitive models perfectly reproduce the homogeneous behavior for *-able* and *-ish*, the type-

Table 4. Human evaluation

	MGL		GCM		LLMs	
Suffix	Type	Token	Type	Token	GPT-J	GPT-4
-able	1.000	1.000	1.000	1.000	0.893	0.960
-ish	1.000	1.000	1.000	1.000	0.997	1.000
-ive	0.720	0.680	0.760	0.700	0.632	0.440
-ous	0.560	0.520	0.640	0.520	0.503	0.400

The table shows the results of rule-based and exemplar-based analogy models as well as GPT-J and GPT-4 evaluated against human annotations. The measure is accuracy.

based GCM model better matches the human predictions for *-ive* and *-ous*, as reflected by large gaps compared to the second-best model, type-level MGL (*-ive*: 4%, *-ous*: 8%). The token-based variants of MGL and GCM match the human behavior substantially worse than the type-based variants, which is exactly in line with what has been suggested in prior work (74, 84). Moving to the results for GPT-J, it turns out to match the human responses worse than any of the cognitive models, for all four adjective classes. The gap compared to the best cognitive model, type-based GCM, is considerable, especially for *-ive* and *-ous*, amounting to roughly 13% in both cases. The picture is overall even worse for GPT-4. While the predictions for the high-regularity classes are good (almost always *-ity* for *-able* and *-ness* for *-ish*, like all other models), the match with humans is more than 10% worse than GPT-J for both *-ive* and *-ous*.

Why do GPT-J and GPT-4 match the human behavior so much worse than the much simpler type-based GCM? The key factor, we argue, is that both of these LLMs are driven by the token frequencies of the words in the training data. Just as for GPT-J, the token-based GCM and MGL models match the behavior of GPT-4 better than the type-based models (*SI Appendix*, Table S3). Token-oriented behavior is desirable in that it results in highly realistic implicit knowledge of individual words, as we have seen above. However, humans step back from the frequencies of individual words when making generalizations about possible words. LLMs seem to lack the ability to do this.

Furthermore, our results suggest that GPT-4's overreliance on token frequency is if anything worse than that of GPT-J. Thus GPT-4's morphological generalization behavior seems to be even less human-like than that of GPT-J. This finding is reminiscent of recently reported "inverse scaling effects," more specifically the tendency of larger LLMs to rely even more strongly on prior statistics from the training data than smaller models do (86).

Since the performance of the best model, the type-based GCM, leaves room for improvement, we can ask why its performance was not better. The single biggest discrepancy was that the GCM selected *-ness* after *-ous* more than the humans did. Our analysis does not deal with the possibility that some people may consider the affix *-osity* to be a unified affix bundle, along the lines suggested in refs. 87 and 88; this would enhance its availability. Given that the GCM was fit to all word pairs attested in the Pile, the analysis also failed to allow for differences among human mental lexicons. Other studies have found considerable variability among human participants in the area of derivational morphology in general, and in preference for *-ity* over *-ness* specifically (89–93). In this context, it is noteworthy that the participants in our study were recruited from a highly educated community, whereas much of the Pile consists of web data such as informal discussions on Reddit (72). There is thus the possibility that there was a misalignment between the sociolect most strongly represented in the Pile and the ideolects sampled as part of our annotation study. Finally, the GCM works on the basis of word forms only, and has no way of taking into account similarities in meaning that also play a role in shaping morphological systems. In contrast, LLMs are able to consider similarities in meaning, but any advantage they might gain from their semantics in this highly focused task appears to be more than offset by the drawbacks of their reliance on token frequencies.

Familiarity of complex words. Our results on nominalizations indicate that GPT-J and GPT-4 do not have a mental lexicon in the sense that humans do, in that they lack the ability to step back from word tokens and generalize over word types. Here, we present a brief demonstration that this observation pertains to morphologically complex words more generally, and not just to nominalizations. For this demonstration, we draw on the Hoosier Lexicon, a dataset of 19,320 English words that includes word frequencies and familiarity ratings on a seven-point Likert Scale (94). An important finding of the original study was a dissociation between word frequency and rated familiarity; one might expect the two to be highly correlated; however, some infrequent words are judged as much more familiar than their frequency would suggest. Needle et al. (95) identify morphological structure as an important factor contributing to this dissociation. A word like *precancellation*, with a recognizable prefix, stem, and suffix seems familiar even though it is rare, on the strength of the familiarity of its parts.

We analyze the $n = 2{,}835$ words in the Hoosier lexicon that have a frequency of less than 10,000 in the Pile (corresponding to a frequency of roughly 1 in 50,000,000 words or less). Leveraging the CELEX dictionary (54) and methodology from prior work (96, 97), we use affix-stripping to identify the $n = 1{,}005$ words that exemplify derivational morphology by virtue of being parsable as a simpler word plus any combination of affixes. $n = 1{,}830$ words cannot be parsed in this way, and we consider them to be simplex words (*SI Appendix*, Supporting Text). For human judgments, we take the familiarity ratings reported by Nusbaum et al. (94). We estimate the "familiarity" that GPT-J assigns to a word as the log probability that it assigns in the context of neutral prompts. Comparing log probabilities to human familiarity ratings is justified because the probabilities assigned to words by language models are known to correlate with psycholinguistic measures of lexical access (e.g., reading times; 98), which for humans are impacted by familiarity to a larger extent than frequency (99).

Results for humans are displayed in Fig. 4*A*. The average familiarity of words with a morphological parse ($n = 1{,}005$) is significantly higher than that of words with no morphological parse ($n = 1{,}830$), $t(2{,}120.2) = 19.2$, $P < 0.001$ (Welch's t test). This confirms the results reported by Needle et al. (95).

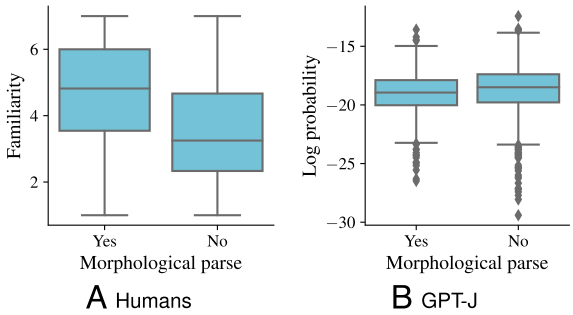


Fig. 4. Impact of morphological decomposability of words on their familiarity as rated by human annotators (*A*) and the log probability assigned to them by GPT-J (*B*). Parsability increases familiarity for humans (*A*), but not for GPT-J (*B*).

Due to this important factor, the correlation between familiarity and log frequency in the entire Hoosier lexicon proves to be modest according to a linear regression, $F(1, 19{,}318) = 11{,}251.2$, $R^2 = 0.368$, $P < 0.001$. For GPT-J, on the other hand, words with a morphological parse do not have any advantage (Fig. 4*B*); quite the opposite, the estimated familiarity of words with a morphological parse is significantly lower than the estimated familiarity of words with no morphological parse for GPT-J, $t(2{,}285.9) = -4.9$, $P < 0.001$. This outcome can be explained by the fact that the correlation between the log frequencies and the log probabilities assigned to the words by GPT-J is very high, $F(1, 19{,}318) = 58{,}553.5$, $R^2 = 0.752$, $P < 0.001$, and the target words without a parse have somewhat higher average frequency ($m = 4{,}285.1$) than those having a parse ($m = 4{,}093.7$).

Experimental studies on wordlikeness judgments (95) and on speech perception (100–102) show that humans continually monitor for known words inside rare or novel words. This means that their type-level lexical representations are exploited during processing, and can cause rare words to seem familiar. GPT-J does not rely on this type-level mechanism and hence lacks the dissociation between frequency and familiarity that is caused by morphological structure.

Discussion

This paper provides empirical evidence for analogical linguistic generalization in LLMs. We found that an analogical cognitive model best explains how GPT-J nominalizes unseen nonce adjectives whose adjective class exhibits a high degree of variability. While the analogical and the rule-based model explained GPT-J's predictions on adjective classes with a high degree of regularity equally well, we showed that the rule-like behavior for those adjectives is the end point of a continuum, where the position of an adjective class is precisely predictive from the level of heterogeneity in the training data. This result is in line with the predictions made by exemplar-based analogy models (e.g., ref. 25). It is not consistent with assuming rules, even with rules having a limited role, as in dual-mechanism approaches. We further found that GPT-J stores a considerable quantity of seen derivatives in its weights, again in line with analogical generalization.

Humans have been also argued to employ analogical generalization in adjective nominalization. However, while humans generalize based on types, we showed that GPT-J generalizes based on tokens. Is this difference between humans and LLMs reflected by their predictions? Indeed it is: The predictions of GPT-J, and similarly of GPT-4, are less human-like than any of the examined cognitive models. We further found that a central manifestation of type-level representations in humans, the decomposition of complex words into morpheme types, is not mirrored by language models. This suggests a critical difference in the organization of the lexicon between humans and LLMs. While humans have a mental lexicon organized around types, the lexical knowledge of LLMs is organized around tokens. Given that analogical generalization mechanisms depend on the lexicon they are operating on, a non-human-like lexicon leads to non-human-like generalizations.

We have presented an intensive study of a single morphological process, nominalization. However, our study has broader repercussions for the language sciences. In historical linguistics, a common trend is for the irregular forms in an inflectional paradigm to become regularized over time; typically, rare forms are affected first (83, 103, 104). However, in some cases, neologisms in a language take on the irregular form, and regular forms can even shift to become irregular (81, 105). In theories of language change, these phenomena are discussed under the rubric of analogical pressure; it is assumed that the acquisition, memory, or production of any inflected form is influenced by pressure from related forms in its neighborhood. Our exploration of frequency effects and neighborhood effects has shown that LLMs can and do operationalize analogical pressure. The boundary between morphology and syntax can be unclear, and in usage-based theories of linguistics, it is also proposed that multiword expressions can be stored in the mental lexicon (106). Krott (107) proposes an analogical approach to productive compounding, and our results suggest that an analogical theory implemented with a deep learning model could also hold promise for documented cases of variability in syntactic constructions (e.g., ref. 58).

Our findings are also related to the ongoing debate about how human-like the language skills of LLMs are (108–110), by highlighting a clear example of an area where the generalizations of LLMs—even the most performant ones—are decidedly non-human-like. The specific shortcoming that we observe in LLMs is that they do not distill token occurrences in text into more abstract type-level representations. From a semantics perspective, this can be interpreted as a failure to semantically ascend (111) to a level of representations that would make it possible for the LLMs to generalize over linguistic objects (specifically, word relations). More generally, this finding can be connected to converging evidence that LLMs fail to form meta-level representations the way humans do (112), in our case meta-level linguistic representations.

Materials and Methods

Cognitive Models. The MGL (73, 74) works by inferring abstract rules from the lexicon. It starts by iterating over pairs of words and forming initial generalizations based on shared phonological features, which are then iteratively merged, yielding increasingly abstract rules. Each rule is associated with a value signifying its statistical reliability. The reliability is derived from the rate at which the rule applies to the *n* forms matching its structural description, adjusted for the uncertainty in the estimate of this rate due to the sample size *n*. To make predictions for a new input, the rule with the highest reliability that matches the phonological properties of the input is selected. In *SI Appendix*, Supporting Text, we provide example rules, including some induced by the MGL.

The GCM (29, 30, 75) does not infer abstract rules but instead stores all forms from the training data in an inventory. To make predictions for a new input, the input is compared to all instances that exhibit each relevant type of output (e.g., to all bases that have a derivative with *-ity*, vs. all bases that have a derivative with *-ness*). The selection of the output pattern is a cumulative effect of similarity and frequency (e.g., the number of different examples weighted by the similarity of those examples to the input). Because of the similarity-based weighting, a small number of highly similar examples can dominate a large number of less similar examples in the decision.

We use the implementation of MGL made available by Albright and Hayes (74), using default hyperparameters. For GCM, our implementation exactly follows prior studies in linguistics using the model (e.g., refs. 24, 27, and 74).

Nonce Adjectives. To create the nonce adjectives for the four adjective classes, we draw upon UniPseudo (76). UniPseudo uses an algorithm based on Markov chains of orthographic *n*-grams that it applies to a specifiable list of input words, generating a list of pseudowords. Importantly, when all input words end in a certain sequence of characters, the generated pseudowords also end in that sequence of characters. We leverage this property of UniPseudo to generate 50 nonce adjectives for each adjective class based on a curated list of adjectives drawn from CELEX (113), MorphoLex (114), and MorphoQuantics (115). For pseudoword length, we use the two most frequent lengths as measured on the extracted adjectives for each class and generate 25 pseudowords for each length. We use the bigram algorithm. See *SI Appendix*, Table S1 for the full list of pseudowords.

GPT-J. We describe the method we use to probe GPT-J more formally. Let b be a base (e.g., *sensitive*) and s be a suffix (e.g., *-ity*). We denote with $d(b, s)$ the derivative resulting from adding s to b and applying all required morpho-orthographic changes (e.g., deletion of base-final e). For instance, for $b = sensitive$ and $s = \text{-}ity$, we have $d(b, s) = sensitivity$. To measure the probability that GPT-J assigns to $d(b, s)$ as a derivative of b, we use various prompts $t(b)$. While some of the prompts ask GPT-J to nominalize b (e.g., $t(b) = Turn\ the\ given\ adjective\ into\ a\ noun.\ b \ \rightarrow$), others are less explicit (e.g., $t(b) = b \ \rightarrow$). See *SI Appendix*, Supporting Text for the full set of prompts.

Given a filled prompt $t(b)$, we pass it through GPT-J and measure the probability that GPT-J assigns to the two derivatives $d(b, \text{-}ity)$ and $d(b, \text{-}ness)$ as continuations of $t(b)$.

We use the GPT-J implementation available on Hugging Face (116). GPT-J has a total of 6,053,381,344 parameters. All experiments are performed on a stack of eight GeForce GTX 1080 Ti GPUs (11 GB).

Adjective Annotation. For determining whether humans prefer *-ity* vs. *-ness* for the nonce adjectives, we collected human judgments from volunteers using the SoSciSurvey platform. Native speakers of English were recruited in a university community using snowball sampling. Hence most or all of them have university-level education. They were asked whether they were willing to participate in a survey about derivational morphology. They were unaware of the exact goals of the study. In total, 28 participants took part. Responses of six participants were removed because they did not finish the survey (attrition rate of 21.4%). Before starting the survey, the participants saw a consent form that described the study and explained that the anonymous responses would be collected, stored, and used for research purposes. They clicked "yes" to indicate their consent and continue on to the survey. The collection and use of the data were submitted to the institutional review board of the Allen Institute for AI for review. It ruled that the study was exempt from regulation because no personally identifiable information would be collected. See *SI Appendix*, Supporting Text for more details about the annotation study.

GPT-4. Since the OpenAI API does not provide access to output probabilities, we cannot use the same method as for GPT-J. Instead, we leverage GPT-4's instruction-following capabilities and directly ask it which of the two derivatives it prefers for a given nonce adjective.

Vocabulary Test. To measure the probability that GPT-J assigns to a word w, we use various prompts $t(w)$ (e.g., $t(w) = Thefollowingisaword\ :w$). See *SI Appendix*, Supporting Text for the full set of prompts.

Given a filled prompt $t(w)$, we pass it through GPT-J and measure the probability that GPT-J assigns to the word.

Data, Materials, and Software Availability. Outputs of computational models have been deposited in GitHub (117). Anonymized data on human judgments of nonwords collected in a crowd-sourcing experiment have been deposited in GitHub (117). The following data cannot be shared: One study reported in the article uses the Hoosier Lexicon dataset described in the study of Nusbaum et al. (94). This heavily cited work predates the FAIR standards by several decades. No online repository ever existed. Research groups using the data received it under a do-not-recirculate agreement. We feel that this situation does not compromise the replicability of our own work, because the study of Nusbaum et al. includes ample detail about how the data were collected, and its main claims have been independently validated by other laboratories. Previously published data were used for this work: CELEX: (113). MorphoLex: (114). MorphoQuantics: (115).

ACKNOWLEDGMENTS. This work was funded by the Engineering and Physical Sciences Research Council (grant EP/T023333/1 awarded to University of Oxford) and the European Research Council (grant 740516 awarded to Ludwig Maximilian University of Munich). V.H. was supported by the German Academic Scholarship Foundation. L.W. was supported by the German Academic Exchange Service. We thank the reviewers for their very helpful comments.

1. J. Hoffmann *et al.*, Training compute-optimal large language models. arXiv [Preprint] (2022). https://doi.org/10.48550/arXiv.2203.15556 (Accessed 20 November 2024).
2. Gemini Team Google, Gemini: A family of highly capable multimodal models. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2312.11805 (Accessed 20 November 2024).
3. OpenAI, GPT-4 Technical report. arXiv [Preprint] (2024). https://doi.org/10.48550/arXiv.2303.08774 (Accessed 20 November 2024).
4. H. Touvron *et al.*, LLaMA: Open and efficient foundation language models. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2302.13971 (Accessed 20 November 2024).
5. A. Q. Jiang *et al.*, Mistral 7B. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2310.06825 (Accessed 20 November 2024).
6. D. Groeneveld *et al.*, OLMo: Accelerating the science of language models. arXiv [Preprint] (2024). https://doi.org/10.48550/arXiv.2402.00838 (Accessed 20 November 2024).
7. A. Chowdhery *et al.*, PaLM: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **24**, 1–113 (2023).
8. R. Dale, GPT-3: What's it good for? *Nat. Lang. Eng.* **27**, 113–118 (2021).
9. H. Haider, Is chat-GPT a grammatically competent informant? lingbuzz [Preprint] (2023). https://lingbuzz.net/lingbuzz/007285 (Accessed 20 November 2024).
10. E. M. Bender, A. Koller, "Climbing towards NLU: On meaning, form, and understanding in the age of data" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, J. Tetreault, Eds. (Association for Computational Linguistics, 2020), pp. 5185–5198.
11. V. Dentella, E. Murphy, G. Marcus, E. Leivada, Testing AI performance on less frequent aspects of language reveals insensitivity to underlying meaning. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2302.12313 (Accessed 20 November 2024).
12. R. Katzir, Why large language models are poor theories of human linguistic cognition: A reply to Piantadosi. *Biolinguistics* **17**, e13153 (2023).
13. L. Weissweiler *et al.*, "Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model" in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, K. Bali, Eds. (Association for Computational Linguistics, Singapore, 2023), pp. 6508–6524.
14. K. Gulordava, P. Bojanowski, E. Grave, T. Linzen, M. Baroni, "Colorless green recurrent networks dream hierarchically" in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Walker, H. Ji, A. Stent, Eds. (Association for Computational Linguistics, New Orleans, LA, 2018), pp. 1195–1205.
15. C. Haley, "This is a BERT. Now there are several of them. Can they generalize to novel words?" in *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, A. Alishahi *et al.*, Eds. (Association for Computational Linguistics, 2020), pp. 333–341.
16. N. Kim, P. Smolensky, "Testing for grammatical category abstraction in neural language models" in *Proceedings of the Society for Computation in Linguistics 2021*, A. Ettinger, E. Pavlick, B. Prickett, Eds. (Association for Computational Linguistics, 2021), pp. 467–470.
17. R. H. Maudslay, R. Cotterell, "Do syntactic probes probe syntax? Experiments with jabberwocky probing" in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova *et al.*, Eds. (Association for Computational Linguistics, 2021), pp. 124–131.
18. J. Wei, D. Garrette, T. Linzen, E. Pavlick, "Frequency effects on syntactic rule learning in transformers" in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, S. W. Yih, Eds. (Association for Computational Linguistics, Punta Cana, 2021), pp. 932–948.
19. R. T. McCoy, P. Smolensky, T. Linzen, J. Gao, A. Celikyilmaz, How much do language models copy from their training data? Evaluating linguistic novelty in text generation using RAVEN. *Trans. Assoc. Comput. Linguist.* **11**, 652–670 (2023).
20. N. Chomsky, *Aspects of the Theory of Syntax* (MIT Press, Cambridge, MA, 1965).
21. N. Chomsky, M. Halle, *The Sound Pattern of English* (Harper & Row, New York, NY, 1968).
22. R. Skousen, *Analogical Modeling of Language* (Kluwer, Dordrecht, 1989).
23. K. Johnson, "Speech perception without speaker normalization: An exemplar model" in *Talker Variability in Speech Processing*, K. Johnson, J. W. Mullennix, Eds. (Academic Press, San Diego, CA, 1997), pp. 145–165.
24. L. Dawdy-Hesterberg, J. Pierrehumbert, Learnability and generalisation of Arabic broken plural nouns. *Lang. Cogn. Neurosci.* **29**, 1268–1282 (2014).
25. S. Todd, J. Pierrehumbert, J. Hay, Word frequency effects in sound change as a consequence of perceptual asymmetries: An exemplar-based model. *Cognition* **185**, 1–20 (2019).
26. B. Ambridge, Against stored abstractions: A radical exemplar model of language acquisition. *First Lang.* **40**, 509–559 (2020).
27. P. Rácz, C. Beckner, J. B. Hay, J. Pierrehumbert, Morphological convergence as on-line lexical analogy. *Language* **96**, 735–770 (2020).
28. P. Rácz, Á. Lukács, Lexical and social effects on the learning and integration of inflectional morphology. *Cogn. Sci.* **48**, e13483 (2024).
29. R. M. Nosofsky, Attention, similarity, and the identification-categorization relationship. *J. Exp. Psychol. Gen.* **115**, 39–57 (1986).
30. R. M. Nosofsky, Similarity, frequency, and category representations. *J. Exp. Psychol. Learn. Mem. Cogn.* **14**, 54–65 (1988).
31. D. Gentner, K. Holyoak, B. Kokinov, *The Analogical Mind: Perspectives from Cognitive Science* (MIT Press, Cambridge, MA, 2001).
32. J. B. Tenenbaum, T. L. Griffiths, Generalization, similarity, and Bayesian inference. *Behav. Brain Sci.* **24**, 629–640 (2001).

33. S. Biderman *et al.*, Emergent and predictable memorization in large language models. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2304.11158 (Accessed 20 November 2024).
34. B. Cao *et al.*, Retentive or forgetful? Diving into the knowledge memorizing mechanism of language models. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2305.09144 (Accessed 20 November 2024).
35. N. Carlini *et al.*, "Quantifying memorization across neural language models" in *International Conference on Learning Representations 2023* (ICLR, 2023).
36. S. Pinker, A. Prince, On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition* **28**, 73–193 (1988).
37. S. Pinker, A. Prince, "Regular and irregular morphology and the psychological status of rules of grammar" in *Annual Meeting of the Berkeley Linguistics Society*, L. Sutton, C. Johnson, R. Shields, Eds. (Berkeley Linguistics Society, Berkeley, CA, 1991), pp. 230–251.
38. S. Prasada, S. Pinker, Generalisation of regular and irregular morphological patterns. *Lang. Cogn. Process.* **8**, 1–56 (1993).
39. U. Hahn, N. Chater, Similarity and rules: Distinct? Exhaustive? Empirically Distinguishable? *Cognition* **65**, 197–230 (1998).
40. E. M. Pothos, The rules versus similarity distinction. *Behav. Brain Sci.* **28**, 1–14 (2005).
41. S. Arndt-Lappe, "Word-formation and analogy" in *Word-Formation: An International Handbook of the Languages of Europe*, P. O. Müller, I. Ohnheiser, S. Olsen, F. Rainer, Eds. (2015), pp. 822–841.
42. D. E. Rumelhart, J. L. McClelland, On learning the past tenses of English verbs. *Psycholinguist. Crit. Concepts Psychol.* **4**, 216–271 (1986).
43. D. C. Plaut, J. L. McClelland, "Generalization with componential attractors: Word and nonword reading in an attractor network" in *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, M. C. Polson, Ed. (Lawrence Erlbaum Associates, Hillsdale, NJ, 1993), pp. 824–829.
44. D. E. Rumelhart, P. M. Todd, "Learning and connectionist representations" in *Attention and Performance XIV: Synergies in Experimental Psychology, Artificial Intelligence, and Cognitive Neuroscience*, D. E. Meyer, S. Kornblum, Eds. (MIT Press, Cambridge, MA, 1993), pp. 3–30.
45. D. C. Plaut, J. L. McClelland, M. S. Seidenberg, K. Patterson, Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychol. Rev.* **103**, 56–115 (1996).
46. D. C. Plaut, L. M. Gonnerman, Are non-semantic morphological effects incompatible with a distributed connectionist approach to lexical processing? *Lang. Cogn. Process.* **15**, 445–485 (2000).
47. L. M. Gonnerman, M. S. Seidenberg, E. S. Andersen, Graded semantic and phonological similarity effects in priming: Evidence for a distributed connectionist approach to morphology. *J. Exp. Psychol. Gen.* **136**, 323–345 (2007).
48. M. Aronoff, *Word Formation in Generative Grammar* (MIT Press, Cambridge, MA, 1976).
49. L. Bauer, *Morphological Productivity* (Cambridge University Press, Cambridge, UK, 2001).
50. M. Haspelmath, A. Sims, *Understanding Morphology* (Routledge, Oxford, UK, 2010).
51. L. Bauer, R. Lieber, I. Plag, *The Oxford Reference Guide to English Morphology* (Oxford University Press, Oxford, UK, 2013).
52. R. Beard, "Derivation" in *The Handbook of Morphology*, A. Spencer, A .M. Zwicky, Eds. (Blackwell, Oxford, UK, 2017), pp. 44–65.
53. F. Anshen, M. Aronoff, Producing morphologically complex words. *Linguistics* **26**, 641–656 (1988).
54. R. H. Baayen, A. Renouf, Chronicling the times: Productive lexical innovations in an English newspaper. *Language* **72**, 69–96 (1996).
55. F. Anshen, M. Aronoff, Using dictionaries to study the mental lexicon. *Brain Lang.* **68**, 16–26 (1999).
56. M. Lindsay, Rival suffixes: Synonymy, competition, and the emergence of productivity. *Mediterr. Morphol. Meet.* **8**, 192–203 (2012).
57. S. Arndt-Lappe, Analogy in suffix rivalry: The case of English -Ity and -Ness. *Engl. Lang. Linguist.* **18**, 497–548 (2014).
58. J. Bresnan, A. Cueni, T. Nikitina, R. H. Baayen, "Predicting the dative alternation" in *Cognitive Foundations of Interpretation*, G. Bouma, I. Krämer, J. Zwarts, Eds. (Royal Netherlands Academy of Arts and Sciences, Amsterdam, 2007), pp. 69–94.
59. M. Walsh, B. Möbius, T. Wade, H. Schütze, Multilevel exemplar theory. *Cogn. Sci.* **34**, 537–582 (2010).
60. J. Bresnan, Formal grammar, usage probabilities, and auxiliary contraction. *Language* **97**, 108–150 (2021).
61. D. Edmiston, A systematic analysis of morphological content in BERT models for multiple languages. arXiv [Preprint] (2020). https://doi.org/10.48550/arXiv.2004.03032 (Accessed 20 November 2024).
62. V. Hofmann, J. Pierrehumbert, H. Schütze, "DagoBERT: Generating derivational morphology with a pretrained language model" in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, B. Webber, T. Cohn, Y. He, Y. Liu, Eds. (Association for Computational Linguistics, 2020), pp. 3848–3861.
63. V. Hofmann, J. Pierrehumbert, H. Schütze, "Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words" in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, C. Zong, F. Xia, W. Li, R. Navigli, Eds. (Association for Computational Linguistics, 2021), pp. 3594–3608.
64. R. Sun, "Introduction to computational cognitive modeling" in *The Cambridge Handbook of Computational Psychology*, R. Sun, Ed. (Cambridge University Press, Cambridge, UK, 2008), pp. 3–19.
65. R. Wilson, A. Collins, Ten simple rules for the computational modeling of behavioral data. *eLife* **8**, e49547 (2019).
66. A. Brasoveanu, J. Dotlačil, *Computational Cognitive Modeling and Linguistic Theory* (Springer, Cham, 2020).
67. E. Akyurek *et al.*, "Towards tracing knowledge in language models back to the training data" in *Findings of the Association for Computational Linguistics: EMNLP*, Y. Goldberg, Z. Kozareva, Y. Zhang, Eds. (Association for Computational Linguistics, Abu Dhabi, 2022), pp. 2429–2446.
68. X. Han, Y. Tsvetkov, ORCA: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data. arXiv [Preprint] (2022). https://doi.org/10.48550/arXiv.2205.12600 (Accessed 20 November 2024).
69. Y. Razeghi, R. Logan IV, M. Gardner, S. Singh, "Impact of pretraining term frequencies on few-shot numerical reasoning" in *Findings of the Association for Computational Linguistics: EMNLP*, Y. Goldberg, Z. Kozareva, Y. Zhang, Eds. (Association for Computational Linguistics, Abu Dhabi, 2022), pp. 840–854.
70. Y. Elazar *et al.*, Measuring causal effects of data statistics on language model's 'factual' predictions. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2207.14251 (Accessed 20 November 2024).
71. B. Wang, A. Komatsuzaki, GPT-J-6B: A 6 billion parameter autoregressive language model. GitHub. https://github.com/kingoflolz/mesh-transformer-jax. Accessed 1 September 2023.
72. L. Gao *et al.*, The Pile: An 800GB dataset of diverse text for language modeling. arXiv [Preprint] (2020). https://doi.org/10.48550/arXiv.2101.00027 (Accessed 20 November 2024).
73. A. Albright, B. Hayes, "Modeling English past tense intuitions with minimal generalization" in *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, M. Maxwell, Ed. (Association for Computational Linguistics, Philadelphia, PA, 2002), pp. 58–69.
74. A. Albright, B. Hayes, Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition* **90**, 119–161 (2003).
75. R. M. Nosofsky, Relations between exemplar-similarity and likelihood models of classification. *J. Math. Psychol.* **34**, 393–418 (1990).
76. B. New, J. Bourgin, J. Barra, C. Pallier, UniPseudo: A universal pseudoword generator. *Q. J. Exp. Physiol.* **77**, 278–286 (2024).
77. J. W. Rae *et al.*, Scaling language models: Methods, analysis & insights from training Gopher. arXiv [Preprint] (2022). https://doi.org/10.48550/arXiv.2112.11446 (Accessed 20 November 2024).
78. M. L. Hare, M. Ford, W. D. Marslen-Wilson, "Ambiguity and frequency effects in regular verb inflection" in *Frequency and the Emergence of Linguistic Structure*, J. Bybee, P. Hopper, Eds. (John Benjamins, Amsterdam, 2001), pp. 181–200.
79. S. Arndt-Lappe, M. Ernestus, "Morpho-phonological alternations: The role of lexical storage" in *Word Knowledge and Word Usage: A Cross-Disciplinary Guide to the Mental Lexicon*, V. Pirrelli, I. Plag, W. U. Dressler, Eds. (De Gruyter, Berlin, 2020), pp. 191–227.
80. S. Holm, A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **6**, 65–70 (1979).
81. R. Daland, A. D. Sims, J. Pierrehumbert, "Much ado about nothing: A social network model of Russian paradigmatic gaps" in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, A. Zaenen, A. van den Bosch, Eds. (Association for Computational Linguistics, Prague, 2007), pp. 936–943.
82. A. Petrov, P. H. Torr, A. Bibi, "When do prompting and prefix-tuning work? A theory of capabilities and limitations" in *International Conference on Learning Representations 2024* (ICLR, 2024).
83. J. Bybee, Regular morphology and the lexicon. *Lang. Cogn. Process.* **10**, 425–455 (1995).
84. J. Pierrehumbert, Stochastic phonology. *Glot Int.* **5**, 195–207 (2001).
85. J. Pierrehumbert, Phonetic diversity, statistical learning, and the acquisition of phonology. *Lang. Speech* **46**, 115–154 (2003).
86. I. R. McKenzie *et al.*, Inverse scaling: When bigger isn't better. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2306.09479 (Accessed 20 November 2024).
87. G. Stump, Rule conflation in an inferential-realizational theory of morphotactics. *Acta Linguist. Acad.* **64**, 79–124 (2017).
88. G. Stump, Some sources of apparent gaps in derivational paradigms. *Morphology* **29**, 271–292 (2019).
89. J. Pierrehumbert, "The statistical basis of an unnatural alternation" in *Laboratory Phonology 8*, L. Goldstein, D. H. Whalen, C. T. Best, Eds. (De Gruyter, Berlin, 2006), pp. 81–106.
90. T. Säily, Variation in morphological productivity in the BNC: Sociolinguistic and methodological considerations. *Corpus Linguist. Linguist. Theory* **7**, 119–141 (2011).
91. T. Säily, *Sociolinguistic Variation in English Derivational Productivity: Studies and Methods in Diachronic Corpus Linguistics* (Société Néophilologique de Helsinki, Helsinki, 2014).
92. T. Säily, Sociolinguistic variation in morphological productivity in eighteenth-century English. *Corpus Linguist. Linguist. Theory* **12**, 129–151 (2016).
93. J. Pierrehumbert, Phonological representation: Beyond abstract versus episodic. *Annu. Rev. Linguist.* **2**, 33–52 (2016).
94. H. C. Nusbaum, D. B. Pisoni, C. K. Davis, Sizing up the Hoosier mental lexicon. *Res. Spoken Lang. Process. Rep.* **10**, 357–376 (1984).
95. J. M. Needle, J. Pierrehumbert, J. B. Hay, "Phonotactic and morphological effects in the acceptability of pseudowords" in *Morphological Diversity and Linguistic Cognition*, A. D. Sims, A. Ussishkin, J. Parker, S. Wray, Eds. (Cambridge University Press, Cambridge, UK, 2022), pp. 79–112.
96. V. Hofmann, J. Pierrehumbert, H. Schütze, "Predicting the growth of morphological families from social and linguistic factors" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, J. Tetreault, Eds. (Association for Computational Linguistics, 2020), pp. 7273–7283.
97. V. Hofmann, H. Schütze, J. Pierrehumbert, "A graph auto-encoder model of derivational morphology" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, J. Tetreault, Eds. (Association for Computational Linguistics, 2020), pp. 1127–1138.
98. E. G. Wilcox, T. Pimentel, C. Meister, R. Cotterell, R. P. Levy, Testing the predictions of surprisal theory in 11 languages. *Trans. Assoc. Comput. Linguist.* **11**, 1451–1470 (2023).
99. M. Brysbaert, P. Mandera, S. F. McCormick, E. Keuleers, Word prevalence norms for 62,000 English lemmas. *Behav. Res. Methods* **51**, 467–479 (2019).
100. K. Rastle, M. H. Davis, B. New, The broth in my brother's brothel: morpho-orthographic segmentation in visual word recognition. *Psychon. Bull. Rev.* **11**, 1090–1098 (2004).
101. M. Taft, Morphological decomposition and the reverse base frequency effect. *Q. J. Exp. Physiol.* **57**, 745–765 (2004).
102. E. Beyersmann, J. C. Ziegler, J. Grainger, Differences in the processing of prefixes and suffixes revealed by a letter-search task. *Sci. Stud. Read.* **19**, 360–373 (2015).
103. G. Corbett, A. Hippisley, D. Brown, P. Marriott, "Frequency, regularity, and the paradigm" in *Frequency and the Emergence of Linguistic Structure*, J. Bybee, P. Hopper, Eds. (John Benjamins, Amsterdam, 2001), pp. 201–228.

104. E. Lieberman, J. B. Michel, J. Jackson, T. Tang, M. A. Nowak, Quantifying the revolutionary dynamics of language. *Nature* **449**, 713–716 (2007).
105. A. Wedel, "Resolving pattern conflict: Variation and selection in phonology and morphology" in *Analogy in Grammar*, J. Blevins, J. Blevins, Eds. (Oxford University Press, Oxford, UK, 2009), pp. 83–100.
106. J. Bybee, From usage to grammar: The mind's response to repetition. *Language* **82**, 711–733 (2006).
107. A. Krott, "The role of analogy for compound words" in *Analogy in Grammar*, J. Blevins, J. Blevins, Eds. (Oxford University Press, Oxford, UK, 2009), pp. 118–136.
108. V. Dentella, F. Günther, E. Leivada, Systematic testing of three language models reveals low language accuracy, absence of response stability, and a yes-response bias. *Proc. Natl. Acad. Sci. U.S.A.* **120**, e2309583120 (2023).
109. J. Hu, R. Levy, "Prompting is not a substitute for probability measurements in large language models" in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, K. Bali, Eds. (Association for Computational Linguistics, Singapore, 2023), pp. 5040–5060.
110. J. Hu, K. Mahowald, G. Lupyan, A. Ivanova, R. Levy, Language models align with human judgments on key grammatical constructions. *Proc. Natl. Acad. Sci. U.S.A.* **121**, e2400917121 (2024).
111. W. V. Quine, *Word and Object* (MIT Press, Cambridge, MA, 1960).
112. N. Dziri *et al.*, Faith and fate: Limits of transformers on compositionality. arXiv [Preprint] (2023). https://doi.org/10.48550/arXiv.2305.18654 (Accessed 20 November 2024).
113. R. H. Baayen, R. Piepenbrock, L. Gulikers, *The CELEX Lexical Database (CD-ROM)* (Linguistic Data Consortium, Philadelphia, PA, 1995).
114. C. H. Sánchez-Gutiérrez, H. Mailhot, S. H. Deacon, M. A. Wilson, MorphoLex: A derivational morphological database for 70,000 English words. *Behav. Res. Methods* **50**, 1568–1580 (2018).
115. J. Laws, C. Ryder, Getting the measure of derivational morphology in adult speech: A corpus analysis using morphoquantics. *Lang. Stud. Work. Pap.* **6**, 3–17 (2014).
116. T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing" in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Q. Liu, D. Schlangen, Eds. (Association for Computational Linguistics, 2020), pp. 38–45.
117. V. Hofmann, L. Weissweiler, D. R. Mortensen, H. Schütze, J. B. Pierrehumbert, Derivational morphology reveals analogical generalization in large language models. GitHub. https://github.com/valentinhofmann/llms-adjective-nominalization. Deposited 18 November 2024.

Supporting Information for

Derivational Morphology Reveals Analogical Generalization in Large Language Models

Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, Janet B. Pierrehumbert

To whom correspondence may be addressed. Email: valentinh@allenai.org or janet.pierrehumbert@oerc.ox.ac.uk.

This PDF file includes:

Supporting text
Figs. S1 to S3
Tables S1 to S3
SI References

Supporting Information Text

Example Rules. The standard notation for linguistic rules is as follows:

$$\text{SD} \rightarrow \text{SC} \,/\, \text{LC} \,\_\_\_\, \text{RC}$$

Here, SD is the structural description of the rule, SC specifies the change to produce the output, LC is an optional left-hand context, and RC is an optional right-hand context.

Many researchers might suggest the following default rule for the phonological spell-out of nominalization. NOM represents the underlying morpheme that may be spelled out as *-ness* or *-ity* at the phonological level. (Throughout the paper we make the simplifying assumption that the two spell-outs are synonymous.)

$$\text{NOM} \rightarrow \textit{-ness}$$

This rule has no left or right context because the morpheme NOM would only occur at the morphosyntactic level on stems with suitable syntactic and semantic properties. Under this assumption, all forms in *-ity* would be memorized exceptions. The most statistically reliable rule for our dataset — and one that is induced by the MGL — includes a specification of the left context:

$$\text{NOM} \rightarrow \textit{-ness} \,/\, \left\{\begin{array}{l} \textit{-ed} \\ \textit{-ing} \\ \textit{-ish} \\ \textit{-less} \end{array}\right\} \,\_\_$$

This rule corresponds to the blue dots in Fig. 3 in the main article. The following rule for R-ITY is also induced by the MGL but is less reliable:

$$\text{NOM} \rightarrow \textit{-ity} \,/\, \left\{\begin{array}{l} \textit{-able} \\ \textit{-al} \\ \textit{-ar} \\ \textit{-ic} \end{array}\right\} \,\_\_$$

This rule corresponds to the orange dots in Fig. 3 in the main article.

Prompts. We want to test which of two derivatives — the one ending in *-ness* or the one ending in *-ity* — is preferred by a language model. To do so, we need to measure the probability that the language model assigns to the two competing forms. For example, we need to measure the probability that the language model assigns to *sensitiveness*, and the probability that it assigns to *sensitivity*. Language models such as GPT-J and GPT-4 always assign probabilities to tokens *given a sequence of preceding tokens*. Therefore, in order to measure the probability that a language model assigns to a specific derivative, we need to decide on what tokens to use as the preceding context. This is commonly referred to as *prompting*, and the sequence of preceding tokens that is fed into the language model as *prompt* (1). Properties of the prompt (e.g., the exact wording of a request) can substantially affect the language model predictions (2), which is why it has become common practice to examine several different prompts when analyzing the behavior of language models. Here, we use the following 12 prompts to measure the probabilities that GPT-J assigns to the derivatives:

- *Nominalized adjective:*
- *Noun:*
- *The following is a nominalized adjective:*
- *The following is a noun:*
- *b* →
- *b :*
- *b -*
- *b*
- *Adjective: b Nominalization:*
- *Form the nominalization of the given adjective. b* →
- *Nominalize the given adjective. b* →
- *Turn the given adjective into a noun. b* →

As in the main text, *b* here is a variable that refers to a base. For example, with the prompt *Nominalized adjective:* and the base *sensitive*, we measure the probability assigned to *sensitivity* in the context *Nominalized adjective: sensitivity* as well as the probability assigned to *sensitiveness* in the context *Nominalized adjective: sensitiveness.* The presented results are averaged across prompts; for example, to get GPT-J's match with the cognitive models, we calculate the match based on each of the 12 prompts and report the mean of these 12 scores.

We use the following prompts to measure the probabilities that GPT-J assigns to the words in the vocabulary test:

- *Word:*
- *Real word:*
- *The following is a word:*
- *The following is a real word:*

Derivative Statistics. We analyze the statistics of *-ity* and *-ness* derivatives in the Pile (Table S2). We first focus on *type frequency*, i.e., the number of different derivatives contained in the Pile. For most classes, there is a clear preference for either *-ity* or *-ness*, the only two exceptions being adjectives ending in *-ive* and *-ous*. For adjectives ending in the Germanic suffixes *-ed*, *-ing*, *-ish*, and *-less*, there is a particularly strong preference for *-ness*, although a few derivatives in *-ity* can be found in the data. These statistics are similar to the results of a recent analysis based on dictionary data (3), indicating that the Pile provides a realistic picture of the variation between *-ity* and *-ness* in present-day English.

Next, we turn to *token frequency*, i.e., the number of times individual derivatives occur in the Pile. We notice that the trends for type frequency are largely reflected by token frequency: in the case of adjective classes for which *-ity* derivatives have a higher type frequency than *-ness* derivatives, *-ity* derivatives also tend to have a higher average token frequency than *-ness* derivatives (and vice versa). The only exception is *-ous*, where *-ity* has a lower type frequency but a higher average token frequency than *-ness*. This is due to a particularly large number of *-ity* derivatives in the high token frequency range: excluding the top 5% of derivatives with the highest token frequency, the average token frequency is higher for *-ness* (73.0) than *-ity* (15.1), in line with the type frequency trend for *-ous*.

Finally, we examine a measure that linguistic scholarship has suggested to be particularly relevant for productivity (4, 5), specifically the number of *hapaxes* (i.e., derivatives occurring only once in the Pile). Here, the trends for individual adjective classes are similar to type frequency and token frequency, with the potential exception of *-ive*, where the preponderance of *-ity* compared to *-ness* is slightly less pronounced.

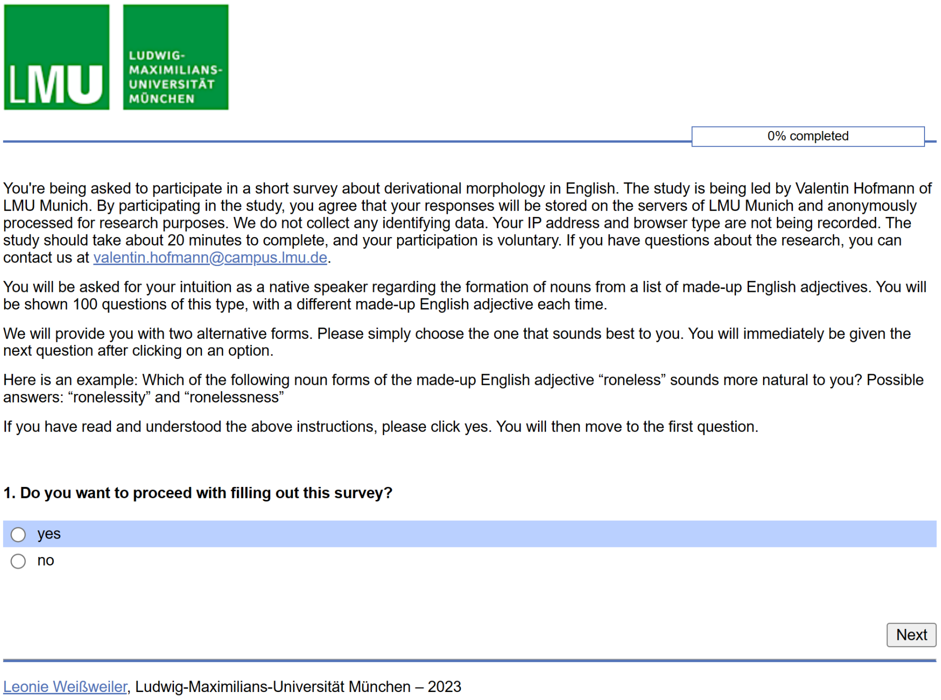
Adjective Annotation. Each participant coded half of the nonce words (i.e., 100 nonce words). The 22 participants who completed the full survey were evenly divided between the two halves. Participants were first shown an introductory message explaining the task as shown in Fig. S1a. Participants who consented to the collection and use of their data, as described in the introductory message, indicated their consent by clicking "yes". They were then given one of two survey versions, each with 100 nonce words that cycled through the four suffixes to avoid repetition. To reduce the total time necessary for completing the survey, participants were immediately shown the next question upon clicking a word. An example of a question is shown in Fig. S1b.

Fig. S2b plots for each tested adjective class the ratio of bases for which participants overall preferred *-ness* over *-ity*, i.e., more participants selected the *-ness* rather than the *-ity* derivative. There is a clear preference for *-ity* in the case of *-able* and a clear preference for *-ness* in the case of *-ish*. For the two suffixes with a larger degree of competition, *-ive* shows the expected pattern, with participants preferring *-ity* over *-ness* for the majority of bases, but *-ous* shows a preference for *-ity*, which is different from its greater association with *-ness* in the Pile. This can also be seen from the ratio of participants preferring *-ness* over *-ity* for individual bases (see Fig. S3a), which is on average smaller than 50% for *-able* (17.7%), *-ive* (39.8%), and *-ous* (47.5%), and greater than 50% only for *-ish* (95.1%). Fig. S3a also shows a high degree of variation between individual bases of a certain adjective class: e.g., for *-ous*, there are bases for which participants clearly preferred *-ity* (e.g., 81.8% preferred *-ity* for *indaminous*), but there is also a base for which participants exclusively selected *-ness* (100% preferred *-ness* for *rebelorous*).

Participants differed in terms of how often they selected *-ity* or *-ness* for each adjective class (see Fig. S3b). For example, 13 participants preferred *-ity* for *-ous* bases, but nine participants preferred *-ness*. This high degree of variation is reflected by a small inter-annotator agreement (IAA) of 0.335, measured using Fleiss' κ. However, measuring IAA on all bases hides the fact that IAA is substantially higher for *-ish* (0.899) and *-able* (0.587) than for *-ive* (0.096) and *-ous* (0.054), measured using Gwet's AC1 (6). There is also a correlation between the responses given by individual participants for bases of different adjective classes, especially between *-able* and *-ive* (0.417), and *-ive* and *-ous* (0.415), measured using Pearson's r.

Morphological Parse. The parsability of words in the Hoosier lexicon is determined as follows. In a first step, we check whether a word is contained in CELEX (7), a lexical database that contains information about the morphological status of more than 50,000 English words. 16,417 words from the Hoosier lexicon are listed in CELEX. For the remaining 2,903 words, we determine the morphological status by means of a simple method from prior work (8, 9): we test whether the beginning or end of words matches common prefixes/suffixes of the English language, and whether the remaining part of the word is a stem. To do so, we draw upon a list of 46 English prefixes and 44 English suffixes (10). As potential stems, we use all English words contained in CELEX. The algorithm is sensitive to morpho-orthographic rules of English (11).

As a result of this procedure, 6,499 words from the Hoosier lexicon are classified as morphologically complex. The words are diverse in terms of the involved affixes: except for *pseudo* and *mini*, all affixes from the list mentioned above show up.



You're being asked to participate in a short survey about derivational morphology in English. The study is being led by Valentin Hofmann of LMU Munich. By participating in the study, you agree that your responses will be stored on the servers of LMU Munich and anonymously processed for research purposes. We do not collect any identifying data. Your IP address and browser type are not being recorded. The study should take about 20 minutes to complete, and your participation is voluntary. If you have questions about the research, you can contact us at valentin.hofmann@campus.lmu.de.

You will be asked for your intuition as a native speaker regarding the formation of nouns from a list of made-up English adjectives. You will be shown 100 questions of this type, with a different made-up English adjective each time.

We will provide you with two alternative forms. Please simply choose the one that sounds best to you. You will immediately be given the next question after clicking on an option.

Here is an example: Which of the following noun forms of the made-up English adjective “roneless” sounds more natural to you? Possible answers: “ronelessity” and “ronelessness”

If you have read and understood the above instructions, please click yes. You will then move to the first question.

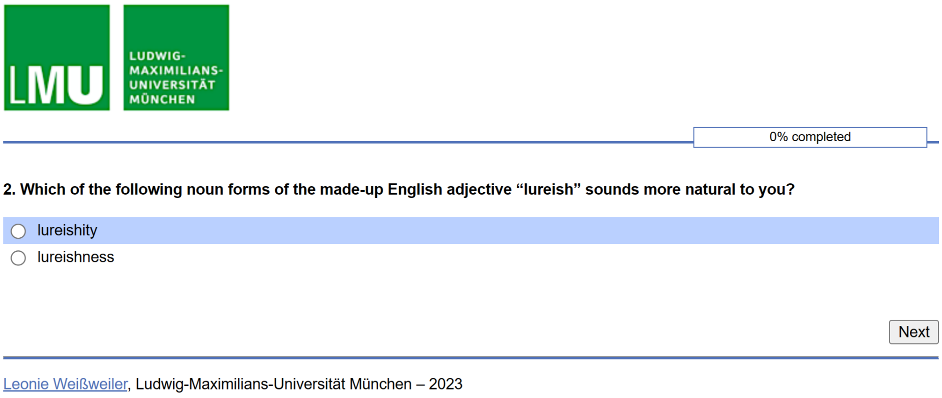
1. Do you want to proceed with filling out this survey?

○ yes

○ no

Leonie Weißweiler, Ludwig-Maximilians-Universität München – 2023

(a) Introductory message



2. Which of the following noun forms of the made-up English adjective “lureish” sounds more natural to you?

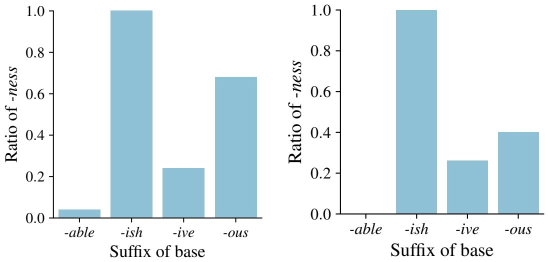
○ lureishity

○ lureishness

Leonie Weißweiler, Ludwig-Maximilians-Universität München – 2023

(b) Example question

Fig. S1. Screenshots of the introductory message seen by participants of our survey (a) and an example question given to participants (b).



(a) GPT-J

(b) Humans

Fig. S2. Distribution of preferred nominalization type (specifically, ratio of *-ness* derivatives) for unseen nonce adjectives, for GPT-J (a) and human annotators (b). The ratio is computed as the number of *-ness* predictions divided by the total number of predictions. Panel (a) replicates Fig. 1e from the main article for easier comparison.

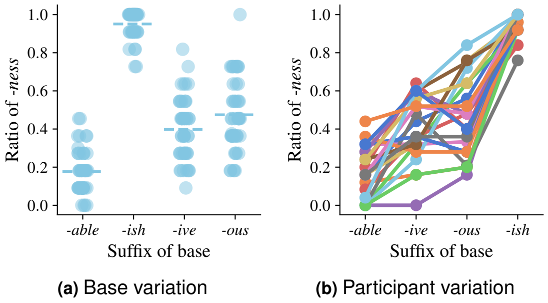


Fig. S3. Variation in the derivative preferred by humans, shown separately for bases (a) and participants (b). In (a), each dot represents one base. In (b), each line represents the response pattern of one participant in our annotation study.

Table S1. Complete list of all used nonce adjectives.

-able	*-ish*	*-ive*	*-ous*
actignable	*badyish*	*atecusive*	*adodagious*
anilicable	*beavish*	*cogective*	*adupendous*
anvastable	*breyish*	*conovative*	*anoninous*
chalinable	*carmish*	*cormasive*	*aurtiguous*
comfolvable	*clangish*	*cuminitive*	*cazardous*
compechable	*clurlish*	*decertive*	*coivonous*
condumable	*cunkish*	*deflosive*	*creninous*
contaitable	*devevish*	*defrertive*	*dardulous*
corgervable	*direish*	*dejovative*	*dexarious*
covornable	*doutish*	*depulsive*	*erenymous*
cresucable	*dwaplish*	*dermasive*	*eretulous*
enocutable	*fadyish*	*dignitive*	*euphitious*
expeaceable	*fawkish*	*dimusitive*	*eutrigeous*
expelocable	*fevetish*	*exhauctive*	*faluminous*
expernable	*fevewish*	*expecative*	*fapturous*
fispoceable	*fevilish*	*extuctive*	*glamalous*
fupeactable	*frietish*	*gederative*	*glumonous*
fusuperable	*friquish*	*imimative*	*gluninous*
imalatable	*ghumpish*	*impuctive*	*gropenious*
impalvable	*gireish*	*indetative*	*hibeguous*
inbeadable	*goguish*	*nogensive*	*honoderous*
inedifiable	*higetish*	*nombasive*	*indaminous*
infoustable	*knarish*	*nonvuptive*	*iniragious*
intoundable	*laretish*	*nutensive*	*insicious*
intountable	*lureish*	*obsensive*	*lasavenous*
inveicable	*lurmish*	*pedititive*	*leamogous*
irediocable	*moguish*	*pedulsive*	*ligegious*
mecoushable	*peftish*	*pepulative*	*liratonous*
parendable	*preanish*	*pransitive*	*luticorous*
peplaicable	*prienish*	*prediasive*	*malicinous*
praleckable	*purerish*	*prititive*	*meglarious*
preneckable	*radyish*	*protrative*	*momogorous*
prequakable	*reckish*	*pumbative*	*mystuorous*
previnable	*redyish*	*recentive*	*nomeneous*
previtable	*rourfish*	*recumotive*	*oblicious*
puneadable	*shigeish*	*rejeptive*	*pecacious*
pustameable	*skierish*	*ruchontive*	*plalorous*
redeptable	*slarish*	*seceptive*	*poncorous*
rempadable	*slownish*	*sejensive*	*prolacious*
retaleable	*slundish*	*serposive*	*ralygerous*
sempoivable	*slungish*	*submiative*	*ravarious*
swimitable	*snoulish*	*submictive*	*reamorous*
tegornable	*sonkish*	*submistive*	*rebelorous*
unaclerable	*tivilish*	*sumpertive*	*slaicitous*
unalintable	*turgeish*	*sumurative*	*suspibious*
undeperable	*wabyish*	*suprective*	*tefigious*
unutintable	*waguish*	*tecensive*	*trospurous*
unvatrable	*wainish*	*tendusive*	*undicitous*
unvediable	*wawkish*	*tredictive*	*vexuteous*
utililable	*woungish*	*vederative*	*vombageous*

Table S2. Statistics of *-ity* and *-ness* derivatives for the 10 examined adjective classes in the Pile (12), the corpus used to train GPT-J (13). The total number of bases is 48,995. The values for token frequency are averaged across all word types belonging to a specific adjective class.

	Type frequency		Token frequency		Hapaxes	
Suffix	*-ity*	*-ness*	*-ity*	*-ness*	*-ity*	*-ness*
-able	11,081	1,034	3937.7	817.3	1,673	226
-al	9,133	1,011	5904.9	172.1	2,078	251
-ar	2,433	214	5833.7	10.3	451	59
-ed	62	4,786	2.4	539.6	28	1,134
-ic	6,215	617	4162.7	45.7	790	175
-ing	2	1,600	1.0	1104.5	2	448
-ish	0	1,502	0.0	397.0	0	437
-ive	4,508	2,438	15075.8	3252.1	626	554
-less	3	2,020	1.7	1159.8	1	506
-ous	1,372	2,450	5453.1	2420.3	325	675

Table S3. Match of rule-based and exemplar-based models with GPT-4 on nonce adjectives.

	MGL		GCM	
Suffix	Type	Token	Type	Token
-able	.960	.960	.960	.960
-ish	1.000	1.000	1.000	1.000
-ive	.400	.480	.440	.500
-ous	.680	.760	.640	.800

References

1. P Liu, et al., Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* **55**, 1–35 (2023).
2. JW Rae, et al., Scaling Language Models: Methods, Analysis & Insights from Training Gopher. Preprint, arXiv 2112.11446 (2022).
3. S Arndt-Lappe, Analogy in Suffix Rivalry: The Case of English -Ity and -Ness. *Engl. Lang. & Linguist.* **18**, 497–548 (2014).
4. H Baayen, R Lieber, Productivity and English Derivation: A Corpus-Based Study. *Linguistics* **29**, 801–844 (1991).
5. RH Baayen, A Renouf, Chronicling the Times: Productive Lexical Innovations in an English Newspaper. *Language* **72**, 69–96 (1996).
6. KL Gwet, Computing Inter-Rater Reliability and Its Variance in the Presence of High Agreement. *Br. J. Math. Stat. Psychol.* **61**, 29–48 (2008).
7. RH Baayen, R Piepenbrock, L Gulikers, *The CELEX Lexical Database (CD-ROM).* (Linguistic Data Consortium, Philadelphia, PA), (1995).
8. V Hofmann, J Pierrehumbert, H Schütze, “Predicting the Growth of Morphological Families from Social and Linguistic Factors” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. pp. 7273–7283 (2020).
9. V Hofmann, H Schütze, J Pierrehumbert, “A Graph Auto-Encoder Model of Derivational Morphology” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. pp. 1127–1138 (2020).
10. D Crystal, *The Cambridge Encyclopedia of the English Language.* (Cambridge University Press, Cambridge, UK), (1997).
11. I Plag, *Word-Formation in English.* (Cambridge University Press, Cambridge, UK), (2003).
12. L Gao, et al., The Pile: An 800GB Dataset of Diverse Text for Language Modeling. Preprint, arXiv 2101.00027 (2020).
13. B Wang, A Komatsuzaki, GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax (2021).

Chapter 12

Corresponds to the following publication:

> **Leonie Weissweiler**, Valentin Hofmann, Masoud Jalili Sabet, Hinrich Schütze (2022). CaMEL: Case Marker Extraction without Labels. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5506–5516, Dublin, Ireland. Association for Computational Linguistics (ACL)

Declaration of Co-Authorship: I conceived the original research contribution together with Valentin Hofmann. I implemented the pipeline and regularly discussed it with Valentin Hofmann and Hinrich Schütze. Masoud Jalili Sabet advised me on the use of his word alignment system. I wrote the initial draft of the paper together with Valentin Hofmann, who drafted the introduction and related work. Hinrich Schütze provided feedback and edited the paper.

Research Context The last chapter in this work was written first, and is not concerned with the evaluation or usage of PLMs. Its computational methodology was motivated purely by a linguistic question: using the overlapping case systems of the world's languages, can we induce deep cases? We attempt to solve this problem using a highly parallel corpus and relatively simple NLP tools. It is unlikely that revisiting this paper in 2024 would change this significantly, so it remains an open challenge: for very low-resource languages, LLMs can not currently be used to induce morphology.

CaMEL: Case Marker Extraction without Labels 🐫

Leonie Weissweiler[*], **Valentin Hofmann**[†*], **Masoud Jalili Sabet**[*], **Hinrich Schütze**[*]

[*]Center for Information and Language Processing, LMU Munich
[†]Faculty of Linguistics, University of Oxford
`{weissweiler,masoud}@cis.lmu.de`
`valentin.hofmann@ling-phil.ox.ac.uk`

Abstract

We introduce **CaMEL** (**Ca**se **M**arker **E**xtraction without **L**abels), a novel and challenging task in computational morphology that is especially relevant for low-resource languages. We propose a first model for CaMEL that uses a massively multilingual corpus to extract case markers in 83 languages based only on a noun phrase chunker and an alignment system. To evaluate CaMEL, we automatically construct a silver standard from UniMorph. The case markers extracted by our model can be used to detect and visualise similarities and differences between the case systems of different languages as well as to annotate fine-grained deep cases in languages in which they are not overtly marked.

1 Introduction

What is a case? Linguistic scholarship has shown that there is an intimate relationship between morphological case marking on the one hand and semantic content on the other (see Blake (1994) and Grimm (2011) for overviews). For example, the Latin case marker *-ibus*[1] (Ablative or Dative Plural) can express the semantic category of location. It has been observed that there is a small number of such semantic categories frequently found cross-linguistically (Fillmore, 1968; Jakobson, 1984), which are variously called *case roles* or *deep cases*. Semiotically, the described situation is complicated by the fact that the relationship between case markers and expressed semantic categories is seldom isomorphic, i.e., there is both *case polysemy* (one case, several meanings) and *case homonymy* or *case syncretism* (several cases, one marker) (Baerman, 2009). As illustrated in Figure 1, the Latin Ablative marker *-ibus* can express the semantic category of instrument besides location (case polysemy), and it is also the marker of the Dative Plural expressing a recipient (case syncretism). In addition, there is *case synonymy* (one case, several markers), which further complicates morphosemiotics; e.g., in Latin, *-is* is an alternative marker of the Ablative Plural.

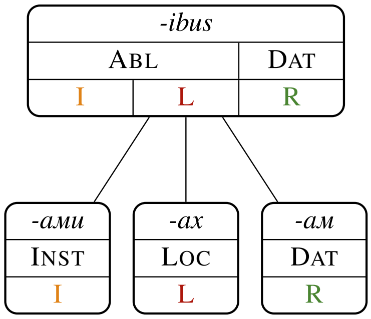


Figure 1: Morpho-semiotic foundation of this study. The Latin case marker *-ibus* is used for both the Ablative (ABL) and the Dative (DAT), which in turn express the three semantic categories of instrument (I), location (L), and recipient (R). This is an example of both case polysemy (one case: ABL, several meanings: I and L) and case syncretism (several cases: ABL and DAT, one marker: *-ibus*). Russian, on the other hand, has an isomorphic relationship between Instrumental (INST), Locative (LOC), and Dative (DAT), the case markers corresponding to them (*-ами*/*-ami*, *-ах*/*-ax*, *-ам*/*-am*), and the expressed semantic categories (I, L, R).

The key idea of this paper is to detect such complex correspondences between case markers and expressed semantic categories in an automated way. Specifically, we build on prior work by Cysouw (2014), who lays the theoretical foundation for our study by showing that deep cases can be induced from cross-linguistic usage patterns of case markers. As opposed to Latin, Russian has separate cases (with separate case markers) for the semantic categories of instrument (*-ами*/*-ami*), location (*-ах*/*-ax*), and recipient (*-ам*/*-am*). Thus, knowing the Russian case marker corresponding to Latin *-ibus* reduces the uncertainty about the expressed

[1] In this paper, we use *italic* when talking about case markers as morphemes in a linguistic context and `monospace` (accompanied by `$` to mark word boundaries) when talking about case markers in the context of our model. Transliterations of Cyrillic examples are given after slashes.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5506 - 5516
May 22-27, 2022 ©2022 Association for Computational Linguistics

case role (Figure 1). This reduction of uncertainty can be particularly helpful in a low-resource setting where other means of analysis are unavailable.

In this work, we rely on the Parallel Bible Corpus (PBC; Mayer and Cysouw, 2014), a massively multilingual corpus, to investigate the relationship between surface cases and their deep meanings cross-linguistically. To put our idea into practice, we require an exhaustive set of case markers as well as a set of parallel noun phrases (NPs) that we can further analyze with respect to deep cases using the set of case markers. Both requirements pose a serious challenge for languages with limited available resources. We therefore introduce **CaMEL** (**Ca**se **M**arker **E**xtraction without **L**abels), a novel and challenging task of finding case markers using only (i) a highly parallel corpus covering many languages, (ii) a noun phrase chunker for English, and (iii) word-level pre-computed alignments across languages.

Our work uses the parallel nature of the data in two ways.

First, we leverage the word-level alignments for the initial step of our pipeline, i.e., the marking of NPs in all languages (even where no noun phrase chunker is available). To do so, we mark NPs in 23 different English versions of the Bible and project these annotations from each English to each non-English version using the word-level alignments, resulting in parallel NPs that express the same semantic content across 83 languages. Based on the projected annotations, we leverage the frequencies of potential case markers inside and outside of NPs as a filter to distinguish case markers from lexical morphemes and other grammatical morphemes typically found outside of NPs.

Second, we leverage the alignments for a fine-grained analysis of the semantic correspondences between case systems of different languages.

We make three main **contributions**.

- We define **CaMEL** (**Ca**se **M**arker **E**xtraction without **L**abels), a new and challenging task with high potential for automated linguistic analysis of cases and their meanings in a multilingual setting.
- We propose a simple method for CaMEL that is efficient, requires no training, and generalises well to low-resource languages.
- We automatically construct a silver standard based on human-annotated data and evaluate our method against it, achieving an F1 of 45%.

To foster future research on CaMEL, we make the silver standard, our code, and the extracted case markers publicly available[2].

2 Related Work

Unsupervised morphology induction has long been a topic of central interest in natural language processing (Yarowsky and Wicentowski, 2000; Goldsmith, 2001; Schone and Jurafsky, 2001; Creutz and Lagus, 2002; Hammarström and Borin, 2011). Recently, unsupervised inflectional paradigm learning has attracted particular interest in the research community (Erdmann et al., 2020; Jin et al., 2020), reflected also by a shared task devoted to the issue (Kann et al., 2020). Our work markedly differs from this line of work in that we are operating on the level of case markers, not full paradigms, and in that we are inducing morphological structure in a massively multilingual setting.

There also have been studies on extracting grammatical information from text by using dependency parsers (Chaudhary et al., 2020; Pratapa et al., 2021) and automatically glossing text (Zhao et al., 2020; Samardžić et al., 2015) as well as compiling full morphological paradigms from it (Moeller et al., 2020). By contrast, our method is independent of such annotation schemata, and it is also simpler as it does not aim at generating full grammatical or morphological descriptions of the languages examined. There has been cross-lingual work in computational morphology before (Snyder and Barzilay, 2008; Cotterell and Heigold, 2017; Malaviya et al., 2018), but not with the objective of inducing inflectional case markers.

Methodologically, our work is most closely related to the SuperPivot model presented by Asgari and Schütze (2017), who investigate the typology of tense in 1,000 languages from the Parallel Bible Corpus (PBC; Mayer and Cysouw, 2014) by projecting tense information from languages that overtly mark it to languages that do not. Based on this, Asgari and Schütze (2017) perform a typological analysis of tense systems in which they use different combinations of tense markers to further divide a single tense in any given language. Our work differs in a number of important ways. First, we do not manually select a feature to investigate

[2] https://github.com/LeonieWeissweiler/CaMEL

but model all features in our chosen sphere of interest (i.e., case) at once. Furthermore, we have access to word-level rather than verse-level alignments and can thus make statements at a more detailed resolution (i.e., about individual NPs). Finally, we extract features not only for a small selection of pivot languages, but even for languages that do not mark case "non-overtly", i.e., in a way that deviates to a large degree from a simple 1–1 mapping (see discussion in §1).

3 Linguistic Background

There is ongoing discussion in linguistic typology about the extent to which syntactic categories are shared and can be compared between the world's languages (see Hartmann et al. (2014) for an overview). While this issue is far from being settled, there is a general consensus that (while not being a language universal) there is a core of semantic categories that are systematically found cross-linguistically, and that are expressed as morphosyntactic case in many languages. Here, we adopt this assumption without any theoretical commitment, drawing upon a minimal set of deep cases detailed in Table 1. The set is loosely based on the classical approach presented by Fillmore (1968).

Going beyond deep cases, Cysouw (2014) envisages a more fine-grained analysis of what is conventionally clustered in a deep case or semantic role. Briefly summarised, the theoretical concept is this: if every language has a slightly different case system, with enough languages it should be possible to divide and cluster NPs at any desired level of granularity, from the conventional case system down to a specific usage of a particular verb in conjunction with only a small set of nouns. For example, the semantic category of location could be further subdivided into specific types of spatial relationships such as 'within', 'over' and 'under'. Taken together, it would then be possible to perform theory-agnostic typological analysis of case-like systems across truly divergent and low-resource languages by simply describing any language's case system in terms of its clustering of very fine-grained semantic roles into larger systems that are overtly marked.

The approach sketched in the last paragraph is not limited to case systems but has been applied to person marking (Cysouw, 2008), the causative/inchoative alternation (Cysouw, 2010), and motion verbs (Wälchli and Cysouw, 2012). The variety of linguistic application areas highlights the potential of developing methods that are much more automated than the work of Cysouw and collaborators. While we stay at the level of traditional deep cases in this paper, we hope to be able to extend our method into the direction of a more general analysis tool in the future.

The remainder of the paper is structured as follows. Section 4 describes our method in detail. Section 5 gives an overview of our results. Finally, Section 6 presents two exploratory analyses.

4 Methodology

4.1 Data

We work with the subset of the PBC (Mayer and Cysouw, 2014) for which the SimAlign alignment algorithm (Jalili Sabet et al., 2020) is available, resulting in 87 languages for our analysis. From the corpus, we only extract those verses that are available in all languages, thus providing for a relatively fair comparison, and remove Malagasy, Georgian, Breton, and Korean, as they have much lower coverage than the other languages. This leaves us with 83 languages and 6,045 verses as our dataset. We also select 23 English versions from the PBC that cover the same 6,045 verses. For each of the 6,045 verses, we then compute $83 \times 23 = 1909$ verse alignments: 83 (for each language) multiplied with 23 (for each English version). In the following, we will describe the components of our pipeline (Figure 2).

4.2 NP Annotation

Because our intermediate goal is to induce complete lists of case markers in all languages we cover, the first step is to restrict the scope of our search to NPs. We hope that this will allow us to retrieve case markers for nouns and adjectives while disregarding verb endings that might otherwise have similar distributional properties. As we are working with 83 languages, most of which are low-resource and lack high-quality noun phrase chunkers, we first identify NPs in English using the spaCy noun phrase chunker (Honnibal et al., 2020) and then project this annotation using the alignments to mark NPs in all other languages. The exception to this are German and Norwegian Bokmål, for which noun phrase chunkers are available directly in spaCy. Because both the spaCy noun phrase chunker and the alignments are prone to error, we make use of 23 distinct English versions

Deep Case	Description	Example
Nominative	The subject of the sentence	He is the Messiah!
Genitive	An entity that possesses another entity	Are you the Judean People's Front?
Recipient	A sentient destination	I gave the gourd to Brian.
Accusative	The direct object of the sentence	Consider the lilies.
Locative	The spatial or temporal position of an entity	They haggle in the market.
Instrumental	The means by which an activity is carried out	The graffiti was written by hand.

Table 1: Descriptions and examples for the deep cases distinguished in this paper, which loosely follow the core deep cases proposed in the classical approach of Fillmore (1968).

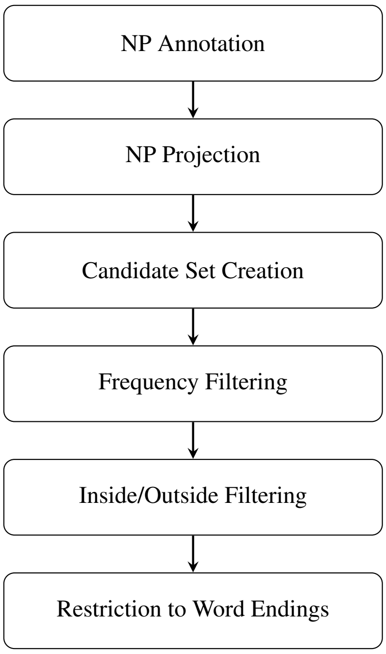


Figure 2: Overview of our pipeline.

of the Bible and mark the NPs in each of them with the goal of lessening the impact of noise.

4.3 NP Projection

We project the NP annotation of a given English version to a second language using the alignments. Specifically, we find the NP in the target language by following the alignments from all words in the English NP while maintaining the word order of the target sentence. We treat each annotated version of the corpus resulting from the different English versions as a separate data source. As an example, Figure 3 shows two English versions and the NP projections for Latin and German. While the alignments, particularly those from English to Latin, are not perfect, they result in complementary errors. The first wrongly aligns the first mention of *pastor bonus*, resulting in only *pastor* being marked as an NP. The second misses the alignment of *life* and *animam*. In these two cases, the other alignment corrects the error.

There are two major results from this process.

First, we obtain the set N of all NPs marked in English, each with all of its translations in the other languages. An example of an entry in this set, taken from Figure 3, would be *the fine shepherd, pastor bonus, der vortreffliche Hirte, ...*, while *the fine shepherd, pastor, der vortreffliche Hirte, ...* would be another, slightly defective, example.

Second, we obtain a pair of multisets, W^l_{in} and W^l_{out}, one for each language l. W^l_{in} (resp. W^l_{out}) is the multiset of all word tokens that appear inside (resp. outside) of NPs of language l. In the following, we will use $M(w)$ to refer to the frequency of word w in the multiset M.

For each language, we want to remove false positives from the word types contained within NPs (which are an artefact of wrong alignments) by using the frequency of each word type inside and outside of NPs.

In principle, this could be done by means of a POS tagger and concentrating on nouns, adjectives, articles, prepositions, and postpositions, but as we do not have access to a reliable POS tagger for most languages covered here, we use the relative frequency information gained from our NP annotations. More specifically, we assign each word type $w \in W^l_{\text{in}} \cup W^l_{\text{out}}$ to I_l (the set of words for language l that are NP-relevant) if $|W^l_{\text{in}}(w)| > |W^l_{\text{out}}(w)|$, and to O_l (the set of words for language l that are not NP-relevant) otherwise. This enhances the robustness of our method against occasional misannotations: for Latin, *ovibus* 'sheep', from our previous example, occurred once outside an NP but 45 times inside and is now an element of I_{Latin}, while *intellegent* 'they understand' occurred once inside an NP but 22 times outside and is therefore an element of O_{Latin}.

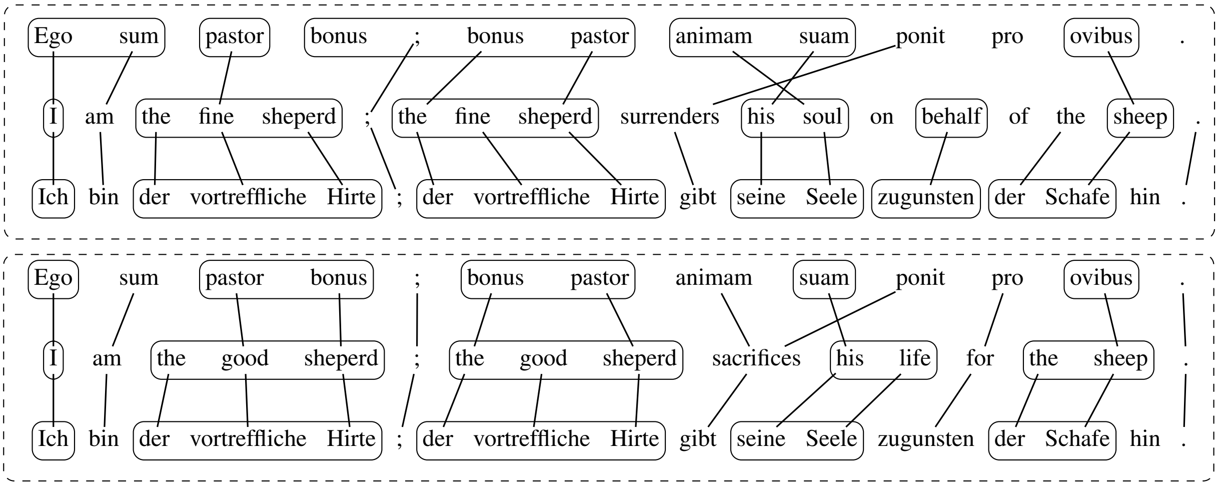

Ego sum pastor bonus ; bonus pastor animam suam ponit pro ovibus .
I am the fine sheperd ; the fine sheperd surrenders his soul on behalf of the sheep .
Ich bin der vortreffliche Hirte ; der vortreffliche Hirte gibt seine Seele zugunsten der Schafe hin .
Ego sum pastor bonus ; bonus pastor animam suam ponit pro ovibus .
I am the good sheperd ; the good sheperd sacrifices his life for the sheep .
Ich bin der vortreffliche Hirte ; der vortreffliche Hirte gibt seine Seele zugunsten der Schafe hin .

Figure 3: Example of alignments and NP projections (English to Latin and English to German) with two different English versions (top and bottom).

4.4 Candidate Set Creation

From each language, we create a set of candidate case markers $\mathrm{candidates}(w)$ for a word w by collecting all character n-grams of any length from w that are also members of I_l. We explicitly mark the word boundaries with `$` so that n-grams in the middle of words are distinct from those at the edges. For example, candidates extracted from *ovibus* would be `$ovi`, `ibus$`, but also `$ovibus$` and `i`. Our first candidate set is computed as $C_1^l = \bigcup\{\mathrm{candidates}(w) \mid w \in I_l\}$.

4.5 Frequency Filtering

We define $I_l(c)$ as the number of words in I_l that contain the candidate c, and $O_l(c)$ analogously for O_l. As a first step, we filter out all n-grams with a frequency in I_l lower than a threshold θ.[3] This results in $C_2^l = \{c \mid c \in C_1^l, I_l(c) \geq \theta\}$.

4.6 Inside/Outside Frequency Filtering

For this step, we make use of the observation that case is a property of nouns. Hence, a case marker is expected to occur much more frequently within NPs. This will serve to distinguish the case markers from verb inflection markers, which should otherwise have similar distributional properties. To implement this basic idea, for each candidate c in language l, we first construct the contingency table shown in Table 2.

	c	$\neg c$
NP	$I_l(c)$	$\sum_{c' \neq c \in C_2^l} I_l(c')$
¬ NP	$O_l(c)$	$\sum_{c' \neq c \in C_2^l} O_l(c')$

Table 2: Contingency table for candidate case marker c in language l for inside/outside filtering. A morphological marker that occurs significantly more often inside NPs than outside of NPs is likely to be a nominal case marker.

We use the table to test whether a candidate is more or less likely to appear inside NPs by comparing the frequencies of the candidate inside and outside NPs to those of all other candidates. Shown in the cells are the frequencies used for the test for each candidate. The columns correspond to the frequency of the candidate in question versus all other candidates while the rows distinguish the frequencies inside versus outside NPs. We carry out a Fisher's Exact Test (Fisher, 1922) on this table, which gives us a p-value and an odds ratio r. $r < 1$ if the candidate is more likely to occur outside an NP, and $r > 1$ if it is more likely to occur inside. The p-value gives us a confidence score to support this ratio (lower is better). We keep for C_{final}^l only those candidates for which $p < \phi$ and $r > \chi$.[4] For example, `ibus$` makes it past this filter with $p(\texttt{ibus\$}) = 2.869 \cdot 10^{-6}$ and $r(\texttt{ibus\$}) = 1.915$ – it is significant and it occurs inside NPs more often than outside NPs. In contrast, `t$` is discarded as it has $p(\texttt{t\$}) = 3.18 \cdot 10^{-149}$ and $r(\texttt{t\$}) = 0.249$ – it is significant, but it has been found to occur much more likely outside than inside NPs.

4.7 Restriction to Word Endings

Suffixoidal inflection is cross-linguistically more common than prefixoidal and infixoidal inflection (Bauer, 2019). This is also reflected in our dataset,

[3] We set $\theta = 97$ based on grid search.

[4] We set $\phi = 0.08$ and $\chi = 0.34$ based on grid search.

where not a single language has prefixoidal or infixoidal inflection. We hence restrict the set of considered n-grams to ones at the end of words.

5 Evaluation of Retrieved Case Markers

We evaluate our method for case marker extraction without labels using a silver standard.

5.1 Silver Standard

As we are, to the best of our knowledge, the first to introduce this task, we cannot rely on an existing set of gold case markers for each language we cover. As most of the languages included are low-resource, reliable grammatical resources do not always exist, which makes the handcrafting of a gold standard difficult. Therefore, and also to ensure relative comparability, we evaluate against a silver standard automatically created from the UniMorph (Sylak-Glassman 2016, Kirov et al. 2018, McCarthy et al. 2020)[5] dataset. The UniMorph data consists of a list of paradigms, which we first filter by their POS tag, keeping only nouns and adjectives and filtering out verbs and adverbs. An example of a paradigm is given in Table 3. While the Nominative Singular (left column) is included in addition to the inflected forms (middle column), the straightforward approach of extracting the suffixes of the inflected forms is not optimal for every language, as the Nominative Singular form can differ from the root. We therefore proceed as follows.

Nominative Singular	inflected forms			unused information
	base		suffix	
Abflug	Abfl	ug		N NOM SG
	Abfl	ug	es	N GEN SG
	Abfl	ug		N DAT SG
	Abfl	ug		N ACC SG
	Abfl	üge		N NOM PL
	Abfl	üge		N GEN PL
	Abfl	ügen		N DAT PL
	Abfl	üge		N ACC PL

Table 3: Example of silver standard creation. Marked in orange is the Nominative Singular form, in red the base ("base") as determined by the algorithm, and in green the only suffix ("suffix") that is extracted from this paradigm. Additional, unused information in the UniMorph data is marked in grey.

First, we form a multiset of all inflected forms. In our example, this would result in {*Abflug, Abfluges, Abflug, Abflug, Abflüge, Abflüge, Abflügen, Abflüge*}. Next, we iterate over this multiset, removing one word each time if it occurs only once. This is meant to make the algorithm more robust against outlier words which do not share a common base with the rest of the paradigm. We then extract the longest common prefix for the remaining elements. We build a frequency list of these prefixes, which in our example has only one element, *Abfl*, with a frequency of 3. We take the most frequent element from the frequency list and compare it to the Nominative Singular, *Abflug*. Of these two candidates, we take the longer one. We thereby prioritise precision over recall as roots that are too short quickly result in many different suffixes that are too long, due to the high overall number of paradigms. Finally, we iterate over the inflected forms again, extracting the suffix if the chosen root is a prefix, which in our example yields one new suffix: `es$`, as *Abflüge* and *Abflügen* are not prefixed by *Abflug*. We examine the results for each language and exclude the languages where either basic knowledge of the language or common sense makes it apparent that sets are much too large or too small, resulting in a diverse set of 19 languages to evaluate our methods against. We note that this process automatically excludes adpositions and clitics, which is in line with our focus on suffixoidal inflection (Section 4.6). We make our silver standard publicly available.

5.2 Results

Our results are provided in Table 4. We observe that precision is higher, at times even substantially, than recall for most languages contained in the silver standard. Looking at Table 5 as an example, we can see that low precision is mostly due to retrieved case markers being longer (`ение$`/*enie*) or shorter (`й$`/*j*) than the correct ones. It is one of the main challenges in this task to select the correct length of a case marker from a series of substring candidates. The shorter substrings will automatically be more frequent and often correct, but this is not easily solved by a frequency threshold, which excludes other correct candidates that are naturally less frequent. Additionally, we observe that some recall errors are due to an incorrect length of n-grams in the silver standard (ьям/*'jam*), highlighting that this issue also exists in its creation process, and suggesting that our performance might even improve when measured against handcrafted data.

[5] https://unimorph.github.io

Language	P	R	F1
Albanian	.74	.47	.58
Belarusian	.43	.41	.42
Bengali	.50	.40	.44
Czech	.50	.58	.54
German	.54	.47	.50
Greek	.67	.19	.30
Icelandic	.83	.31	.45
Indonesian	.31	.42	.36
Irish	.42	.30	.35
Latin	.65	.56	.60
Lithuanian	.18	.38	.24
Nynorsk	.79	.48	.59
Bokmål	.67	.45	.54
Polish	.52	.33	.40
Russian	.54	.54	.54
Slovenian	.41	.28	.33
Swedish	.68	.25	.36
Ukrainian	.45	.48	.47
Average	.54	.41	.45

Table 4: Precision (P), Recall (R) and F1 on the task of case marker extraction without labels for languages contained in our silver standard. Nynorsk and Bokmål are two varieties of Norwegian.

5.3 Ablation Study

We conduct an ablation study to assess the effects of the different pipeline components.

5.3.1 Evaluating NP Projection

In order to evaluate how well our method of projecting NP annotation using alignments to languages without an available NP chunker (see Section 4.3) works, we evaluate it against the monolingual spaCy chunkers for Norwegian Bokmål and German, which are the only available languages besides English. We do not directly compare annotated spans but instead their influence on our method as we have intentionally designed our pipeline to be robust to some noise. As I_l, the set of words considered to be NP-relevant, is the essential output of the annotation projection, we compare two versions, the set as a result of direct NP chunking and the set as a result of our annotation procedure. Taking the former as the ground truth for evaluating the latter (assuming that the directly chunked set has superior quality), we observe an F1 of 88.5 % for German and 67.8 % for Norwegian Bokmål. While these numbers seem low at first, the fact that our overall F1 on Norwegian Bokmål (.54, see Table 4) is better than on German (.50) indicates that the later elements of the pipeline are to a certain extent robust against misclassification of NPs.

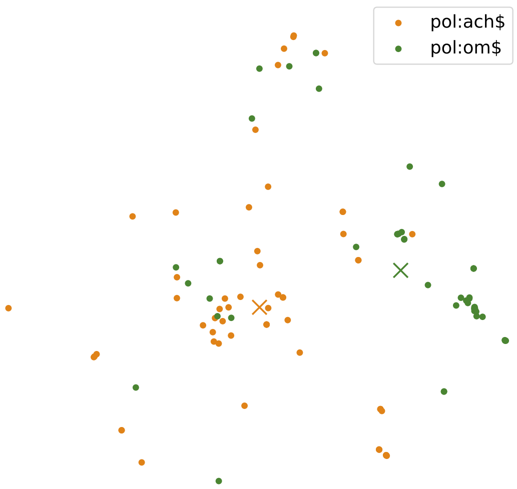


Figure 4: t-SNE plot of the contextual distribution of the Latin case marker *-ibus* and the Polish case markers *-ach* and *-om*. Outliers omitted. The plot shows NPs who in Latin are marked with the case marker `ibus$` and in Polish either with `ach$` (orange) or `om$` (green). Centroids are marked with an X. The plot shows that the Polish case markers exhibit a more fine-grained representation of the underlying semantic categories, which makes it possible to disambiguate the homonymous Latin case marker.

5.3.2 Ablating Pipeline Components

We report the average Precison, Recall, and F1 across all languages in our silver standard without individual filtering components in Table 6. Simple frequency filtering (see "$\neg\theta$"), excluding n-grams within words (see "middle") and at the beginning of words (see "beginning") are all necessary for good performance. Inside/outside filtering based on p-value is the most important component of the pipeline (see "$\neg\phi$"). Surprisingly, inside/outside filtering based on odds ratio has almost no effect.

6 Exploratory Analyses

We can use our automatically extracted case markers, in combination with the parallel NPs that are extracted as part of the pipeline, for innovative linguistic analyses. We present two examples in this section.

6.1 Marking of Deep Cases

First, we demonstrate how, given a parallel NP, the case markers can be used to determine its deep

Intersection	Algorithm Only	Silver Standard Only
у, я, ом, ого, о, в, ой, и, ми, ам, ей, ю, ы, ов, ых, а, м, х, ами	ий, ные, ое, ение, ии, го, ый, ка, ые, к, ки, ия, ние, й, ния, ие	ыми, ах, ев, ьям, ому, ья, н, ьях, ями, ям, е, ях, ьев, ем, ым, ьями
u, ja, om, ogo, o, v, oj, i, mi, am, ej, ju, y, ov, yx, a, m, x, ami	*ij, nye, oe, enie, ii, go, yj, ka, ye, k, ki, ija, nie, j, nija, ie*	*ymi, ax, ev, 'jam, omu, 'ja, n, 'jax, jami, jam, e, jax, 'ev, em, ym, 'jami*

Table 5: The output of our algorithm for Russian compared to the silver standard. We show suffixes that occur in the intersection of algorithm output and silver standard ("Intersection"), those that occur only in the algorithm output ("Algorithm Only") and those that occur only in the silver standard ("Silver Standard Only"). To allow for a clear and concise presentation, the table does not observe the convention of using $ for boundaries.

	ablation	P	R	F1
our method (Table 4)		.54	.41	.45
	$\neg\theta$	.11	.59	.16
	$\neg\phi$	.00	.00	.00
	$\neg\chi$	.53	.41	.44
	middle	.11	.41	.17
	beginning	.33	.41	.35

Table 6: Precision (P), Recall (R) and F1 averaged over all languages on the task of case marker extraction without labels when each step of our pipeline is ablated. $\neg\theta$: no Frequency Filtering; $\neg\phi$: no Inside/Outside Filterung based on p-value; $\neg\chi$: no Inside/Outside Filtering based on odds ratio; middle: include middle of the word; beginning: include beginning of the word.

case. We return to N (see Section 4.3), our set of parallel NPs extracted from the PBC, and for a selected subset of languages, group them by their combination of case markers. The basic idea is to infer an NP's (potentially very fine-grained) deep case by representing it as its combination of case markers across languages.

For example, we can disambiguate the Latin case marker *-ibus* by looking at the different groups the NPs containing it form with Russian case markers. Recall that *-ibus* can express location, instrument, and recipient and that Russian expresses these categories by separate case markers: *-ах*/*-ax* for location, *-ами*/*-ami* for instrument, and *-ам*/*-am* for recipient (see Figure 1) – all three of which have been retrieved by our method. Given a Latin NP marked by the ending *-ibus*, the parallel NP in Russian can help us determine its deep case. Thus, for domibus, дворцах/*dvorcax* shows that the semantic category is location, i.e., 'in the houses'. For operibus bonis, добрыми делами/*dobrymi delami* shows that the semantic category is instrument, i.e., 'through the good deeds'. Finally, for patribus, предкам/*predkam* shows that the semantic category is a recipient, i.e., 'for/to the parents'.

6.2 Similarities between Case Markers

We also demonstrate how we can use their distributional similarities over NPs to show how case markers that are similar in this respect correspond to similar combinations of deep cases. We first generate an NP-word cooccurrence matrix over the NP vocabulary of all languages in which each row, corresponding to an inflected word firm w in language l, indicates which NPs (corresponding to columns) cooccur with w. in the parallel data. We then reduce the dimensionality of the matrix by means of t-SNE (Van der Maaten and Hinton, 2008), allowing us to inspect systematic patterns with respect to the "contexts" in which certain case markers occur (where "context" refers to words the case marker is aligned to in other languages, not words the case marker coccurs with in its own language). In a semiotic situation like the one shown in Figure 1, this setup allows us to examine how the semantic region expressed by a certain homonymous case marker in one language is split into more fine-grained regions in another language that distinguishes the semantic categories that are lumped together by the case marker (and which, if they are at the right level of abstraction, can correspond to deep cases).

Figure 4 shows this scenario for the Latin Ablative marker *-ibus*. It corresponds to two distinct case markers in Polish, *-ach* (LOC) and *-om* (DAT). The figure shows that the region occupied by Latin *-ibus* splits into two distinct clusters in Polish, allowing us to visually determine which underlying case is expressed by the homonymous suffix *-ibus*.

This underscores the exploratory potential of our approach.

7 Conclusion and Future Work

We have introduced the new and challenging task of Case Marker Extraction without Labels (CaMEL) and presented a simple and efficient method that leverages cross-lingual alignments and achieves an F1 of 45% on 19 languages. We introduce an automatically created silver standard to conduct our evaluation. We have further demonstrated two ways in which our retrieved case markers can be used for linguistic analysis.

We see two potential avenues for future work. The first is the further improvement of case marker extraction. The main problem to tackle here is that of small sets of overlapping substrings of which only one is the correct marker, and developing some further measures by which they can be distinguished. Furthermore, it would be useful to find data from more low-resource languages and languages that have typological properties different from the extensively studied large language families (Indo-European, Turkic, Sino-Tibetan etc.). We could then verify that our method performs well across languages and attempt to expand our silver standard to more languages while still ensuring quality. The second area is that of further automating the analysis of deep case and case syncretism. Ideally, we would develop a method that can distinguish the different possible reasons for divergent case marking in languages, with the eventual goal of creating a comprehensive overview of case and declension systems for a large number of languages.

Acknowledgements

This work was funded by the European Research Council (#740516). The second author was also supported by the German Academic Scholarship Foundation and the Arts and Humanities Research Council. The third author was also supported by the German Federal Ministry of Education and Research (BMBF, Grant No. 01IS18036A). We thank the reviewers for their extremely helpful comments.

References

Ehsaneddin Asgari and Hinrich Schütze. 2017. Past, present, future: A computational investigation of the typology of tense in 1000 languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 113–124, Copenhagen, Denmark. Association for Computational Linguistics.

Matthew Baerman. 2009. Case syncretism. In Andrej Malchukov and Andrew Spencer, editors, *The Oxford handbook of case*, pages 219–230. Oxford University Press, Oxford, UK.

Laurie Bauer. 2019. *Rethinking morphology*. Edinburgh University Press, Edinburgh, UK.

Barry J. Blake. 1994. *Case*. Cambridge University Press, Cambridge, UK.

Aditi Chaudhary, Antonios Anastasopoulos, Adithya Pratapa, David R. Mortensen, Zaid Sheikh, Yulia Tsvetkov, and Graham Neubig. 2020. Automatic extraction of rules governing morphological agreement. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5212–5236, Online. Association for Computational Linguistics.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Michael Cysouw. 2008. Building semantic maps: The case of person marking. In *New challenges in typology*, pages 225–248. De Gruyter Mouton.

Michael Cysouw. 2010. Semantic maps as metrics on meaning. *Linguistic Discovery*, 8(1):70–95.

Michael Cysouw. 2014. Inducing semantic roles. *Perspectives on semantic roles*, 23:68.

Alexander Erdmann, Micha Elsner, Shijie Wu, Ryan Cotterell, and Nizar Habash. 2020. The paradigm discovery problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7778–7790, Online. Association for Computational Linguistics.

Charles J. Fillmore. 1968. The case for the case. In Emmon Bach and Robert T. Harms, editors, *Universals in linguistic theory*, pages 1–88. Holt, Rinehart & Winston, New York, NY.

Ronald A Fisher. 1922. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Scott Grimm. 2011. Semantics of case. *Morphology*, 21(3-4):515–544.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Iren Hartmann, Martin Haspelmath, and Michael Cysouw. 2014. Identifying semantic role clusters and alignment types via microrole coexpression tendencies. *Studies in Language*, 38(3):463–484.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Roman Jakobson. 1984. Contribution to the general theory of case: General meanings of the Russian cases. In Linda R. Waugh and Morris Halle, editors, *Russian and Slavic grammar: Studies 1931-1981*, pages 59–103. De Gruyter, Berlin.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Huiming Jin, Liwei Cai, Yihui Peng, Chen Xia, Arya McCarthy, and Katharina Kann. 2020. Unsupervised morphological paradigm completion. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6696–6707, Online. Association for Computational Linguistics.

Katharina Kann, Arya D. McCarthy, Garrett Nicolai, and Mans Hulden. 2020. The SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 51–62, Online. Association for Computational Linguistics.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sabrina J. Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal Morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Chaitanya Malaviya, Matthew R. Gormley, and Graham Neubig. 2018. Neural factor graph models for cross-lingual morphological tagging. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2653–2663, Melbourne, Australia. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020. UniMorph 3.0: Universal Morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association.

Sarah Moeller, Ling Liu, Changbing Yang, Katharina Kann, and Mans Hulden. 2020. IGT2P: From interlinear glossed texts to paradigms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5251–5262, Online. Association for Computational Linguistics.

Adithya Pratapa, Antonios Anastasopoulos, Shruti Rijhwani, Aditi Chaudhary, David R. Mortensen, Graham Neubig, and Yulia Tsvetkov. 2021. Evaluating the morphosyntactic well-formedness of generated texts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7131–7150, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tanja Samardžić, Robert Schikowski, and Sabine Stoll. 2015. Automatic interlinear glossing as two-level sequence classification. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 68–72, Beijing, China. Association for Computational Linguistics.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio. Association for Computational Linguistics.

John Sylak-Glassman. 2016. The composition and use of the universal morphological feature schema (unimorph schema). *Johns Hopkins University*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Bernhard Wälchli and Michael Cysouw. 2012. Lexical typology through similarity semantics: Toward a semantic map of motion verbs. *Linguistics*, 50(3):671–710.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 207–216, Hong Kong. Association for Computational Linguistics.

Xingyuan Zhao, Satoru Ozaki, Antonios Anastasopoulos, Graham Neubig, and Lori Levin. 2020. Automatic interlinear glossing for under-resourced languages leveraging translations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5397–5408, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Part IV

Conclusion and Future Work

In the context of the long debate in Linguistics over the mechanisms underlying human language learning, and the recent advancements in Natural Language Processing, we have explored synergies between Linguistics and modern NLP. These synergies were categorised in two directions: *from Linguistics to NLP*, where we have evaluated pre-trained language models for their linguistic abilities, and *from NLP to Linguistics*, where we proposed applying NLP methods to problems in Linguistics.

This final part is divided into these two research directions, as Chapters 13 and 14. Each is divided into a summary of our answers to the presented research questions, an overview of concurrent related work, and our proposal for future work.

Chapter 13

A More Diverse Assessment of the Linguistic Capabilities of PLMs

The first of our overarching research questions was: How can we diversify the evaluation of the linguistic capabilities of PLMs (**Q1**)?

We now summarise our contributions to answering this question, using Construction Grammar (Section 13.1) and Morphology (Section 13.2) to develop specific investigations. We then summarise concurrent research in similar directions, for both subfields, and give an overview for what we believe to be exciting directions for future research.

13.1 Construction Grammar

Our approach showing how we can diversify the evaluation of the linguistic capabilities of PLMs came from two different sub-fields of Linguistics: Construction Grammar and Morphology.

13.1.1 Our Results

In Section 1.3.1, we asked: 'How well have PLMs learned constructions?". In our work on evaluating PLMs for constructions in Chapters 4 through 7 and Chapter 9, we have shown that even the largest LLMs still struggle with the meaning of some constructions. This was partially due to the design of our studies and does not mean that we can conclude that LLMs don't understand constructions. As we have argued in Chapter 4, from a CxG standpoint, everything is a construction, and therefore LLMs clearly already model many constructions well. The constructions that we have investigated in detail (namely, the English Comparative Correlative, Causal Excess, licensed causal and non-causal, and caused-motion) were chosen intentionally to test the boundaries of LLms, especially with regards to construction meaning. We have found that the boundaries we hypothesised currently exist, as models struggled to apply the construction meaning to novel contexts.

13.1.2 Concurrent Research

During the time that this work was carried out, the idea of evaluating LLMs for their knowledge of constructions has received increased attention. This is evidenced, for example, by the first Construction Grammar and NLP workshop at the Georgetown University Round Table (Bonial and Tayyar Madabushi, 2023), or the two chapters dedicated to the relationship of CxG with Artificial Intelligence and Language Models respectively (Beuls and Eecke, 2025; Madabushi et al., 2025).

Numerous works investigated knowledge of constructions in BERT by looking in detail at the contextual embeddings. Chronis et al. (2023) focused on the Article-Adjective-Numeral-Noun construction (AANN, Goldberg and Michaelis, 2017; Jackendoff, 1977) as well as subject and object roles for noun phrases, and found shifting meaning dimension for key words in the constructions in comparison to outside of it. Veenboer and Bloem (2023) compare BERT's predictions for a masked token with collostructional analysis (Hilpert, 2014; Schmid and Küchenhoff, 2013) of the same token using the BNC (BNC Consortium, 2007) for the "X waiting to happen" (Fillmore et al., 1988) and the ditransitive construction (Goldberg, 1992), and find that the ranked lists of predictions/usages correlate well and constructional information is therefore represented in the model. Tseng et al. (2022) study various Chinese constructions in BERT, and find that it has learned the difference between the open and the closed slots, as its predictions have a higher entropy for the former. Li et al. (2022) study argument structure constructions in RoBERTa, and find evidence that the verbs' contextual embeddings acquire some of the constructional meaning. While this line of work has been creative and enlightening, it is unfortunately impossible with larger autoregressive language models, as the techniques above all rely on a contextual embedding containing both the left and the right context.

As the sole work engaged in using CxG to enhance a PLM, Xu et al. (2023) automatically extract non-overlapping constructions from sentences and use them as the basis of a graph attention network, making constructional information available for downstream tasks. In the era of LLMs, Mahowald (2023) prompts GPT-3 for its grammatical acceptability judgments of the AANN, and finds that it consistently ranks it as grammatical, and slight variations as ungrammatical. Potts (2024) investigate PiPP constructions (Huddleston and Pullum, 2002) and show that the surprisal of GPT-3 given a valid or invalid start of the construction matches human intuition.

From a more linguistic and abstract standpoint, Goldberg (2024) lays out similarities between the learning mechanisms and behaviour of ChatGPT and Construction Grammar, and in particular emphasises the parallels between instruction tuning and the evolutionary pressure on human communication to be helpful.

13.2 Morphology

13.2.1 Our Results

In Section 1.4.1, we asked: "Have PLMs acquired a human-like capacity for morphological generalisation?". Testing the boundaries of LLMs with regards to Morphology in Chapters 10 and 11, we have investigated their capabilities to form human-like morphological generalisations to nonce words. Testing GPT-3.5 on inflectional morphology in Chapter 10, we found it to struggle even in few-shot settings for English and German, but doubly so for Turkish and Tamil. In repeating the experiment for GPT-4[1], we found it to have reached perfect acc@5 for English, for zero-shot and few-shot settings, but worse than GPT-3.5 on the one-shot setting due to over-generalisation to the given example. Evaluating GPT-J and GPT-4 on derivational morphology in Chapter 11, we find that both fail to match human judgements on nonce words. We investigated the errors in more detail and found an overreliance on token, rather than type, frequency, and suboptimal tokenisation. Crucially, we also find evidence that GPT-J represents a continuum between rule-like and analogy-based behaviour, which is fully explainable by the level of heterogeneity in the data. This suggests that the underlying mechanism is a similarity operation on stored exemplars.

13.2.2 Concurrent Research

To the best of our knowledge, no other work has evaluated the morphological capabilities of LLMs.

Most related to our work on the role of analogy in the learning mechanisms of LLMs, Kim and Smolensky (2021) use wug words in contexts that assign them a certain part of speech and investigate BERT's contextual embeddings for them. They find that BERT needs repeated exposure to the novel context to infer the category correctly. Delving deeper into the learning process, Misra and Kim (2023) find that in a similar setup, the representations of the nonce words move consistently towards category exemplars, independent of their initialisation. This might indicate that PLMs form exemplar-based abstract categories for parts of speech.

13.3 General Direction of the Field

Outside of our two more specific subareas, the larger debate in Linguistics and NLP about the suitability of PLMs as models of human linguistic capability has gained both traction and publicity. In an Op-Ed in the New York Times soon after the release of ChatGPT, Chomsky et al. (2023) is of the opinion that "given the [...] linguistic incompetence of these systems, we can only laugh or cry

[1] These results were not published and are available only on the poster that was presented at EMNLP 2023.

at their popularity". The structure of his argument serves as confirmation to our warning in Section 1.2 that the choice of linguistic theory largely decides the outcome of our evaluation of PLMs, as Chomsky bases his criticism on the assumption that "the human mind is a surprisingly efficient and even elegant system that operates with small amounts of information; it seeks not to infer brute correlations among data points".

The opposite standpoint is taken by Piantadosi, who in a preprint entitled "Modern language models refute Chomsky's approach to language" claims that because LLMs work, and because their working principles contradict Chomsky's assumptions, they are proof against his theories. He agrees with Baroni (2022) that language models should be treated as bona fide linguistic theories, or rather as a restriction of possible theories down to those which are consistent with the language models. These bold claims were soon followed by a Chomskyist opinion piece (Katzir, 2023) arguing that Piantadosi is wrong and that LLMs are in facto poor theories of linguistic cognition, and claiming that this will not change with more data and bigger models.

Most recently, Chomsky's claim that LLMs are "incapable of distinguishing the possible from the impossible" has been criticised using practical methods by Kallini et al. (2024), who train GPT-2 variants on artificially constructed "impossible" languages. They find that GPT-2 struggles to learn impossible languages as compared to English, challenging Chomsky's claim.

The experimental setup of studies arguing for or against the linguistic suitability of LLMs remains an issue. Dentella et al. (2023) prompted GPT-3 and GPT-3.5 for grammaticality judgments on a variety of minimal pairs for different grammatical phenomena, and find the responses both inaccurate and biased towards generically answering "yes", which they attribute to Chomsky's claim that probability is fundamentally useless for the notion of grammatical acceptability (Chomsky, 1957). In a response, Hu and Levy (2023) re-examine the data, but use perplexity measures over sentences instead of explicitly prompting the model. With this method, which they have previously shown yields persistently better results on grammatical acceptability, they show that GPT-3.5's judgements are highly correlated with those of humans. Their work also includes an anti-Chomskyist shift in perspective: while Dentella et al. (2023) framed their human results on the dataset as imperfect because of performance issues in humans, Hu and Levy (2023) reframe the human responses as graded acceptability judgements and correlate them with the graded judgements of the model.

As the linguistic competence of LLMs undoubtedly increases, the issue of training data for cognitive plausibility moves into focus. As we have argued in Section 1.1, even a perfect LLM may still be criticised for having been trained on cognitively implausible amounts of data. The BabyLM challenge (Warstadt et al., 2023) was newly initiated as a first step towards alleviating this issue, and presents a corpus of 100M words, which is equivalent to the average lifetime input to a 13-year-old native speaker of English. The shared task then consists of building a language model using only this corpus, which is aimed at challenging participants to develop novel architectures that may be more cognitively plausible. While this is a laudable first effort, the best model won mostly by training for 300 epochs, highlighting the difficulty of defining exact parameters that would make both the data and the training setup as cognitively plausible as possible, while still allowing for creative ideas and architectures.

13.4 Future Work

We now discuss exciting avenues for future work in the area of evaluating LLMs for their linguistic capabilities in comparison with humans. Notably, this section is not divided into CxG and Morphology, as we hope to combine aspects of both in future research.

13.4.1 Construction Meaning

It has become clear that while LLMs seem to have no difficulty recognising and producing the syntactic aspect of constructions, our research and that of others have shown that they struggle with semantically complex constructions. We believe that investigating the causes for this discrepancy could yield insights not only into the inner workings of LLMs, but also show us potential defects in the architecture or training setup.

The essential question is therefore: what causes this issue? Is it a lack of data for very rare constructions, or a lack of diversity in the data (possibly relating to genres with non-creative and repetitive usages rather than diverse and novel ones)? Is it simply the size of the model and will a larger model have the representational capacity to solve this? Or is it, as some have suggested (Gu and Dao, 2024; Zador et al., 2023), a flaw in the architecture hindering its ability to generalise and therefore acquire the semantic nuances of constructions?

These questions are also connected to the notion of compositionality, as it has frequently been argued that LLMs are not compositional enough (Smolensky et al., 2022). In contrast, we have found that the most challenging constructions for LLMs have been those with non-compositional meaning, perhaps indicating that the models are too compositional in some sense. Or instead, they could be struggling on both fronts, and not appropriately adjust the level of compositionality as needed.

To study this in more detail, we believe that a more systematic setup is needed, which is made possible by recent advances in open language models and training data.

The first step is better *knowledge of the conditions* of the experiment. Most simply, this can mean access to the training data of the models we investigate, which has been difficult so far because many models do not disclose their exact training data. However, two models are now available where this is not the case: GPT-J [2], trained on the Pile, and OLMo (Groeneveld et al., 2024), trained on Dolma (Soldaini et al., 2024). Using these models, we hope to be able to count the occurrences of the constructions that we are investigating in the training data, and test correlations between data frequencies and the model's performance. This amounts to a combination of our CxG work with the methodologies introduced in Chapter 11.

The second step is to also have access to the internal states of the model, as is the case for GPT-J and OLMo, but also for Mistral models (Jiang et al., 2023) and the recently announced Gemma models.[3] This will enable us to directly access the perplexities and make clearer statements than by using prompting, as advised by Hu and Levy (2023).

The third ideal step is to have access to models in different sizes, which is the case for the OPT suite of models as well as for Mistral and Gemma, and will soon be the case for OLMo.

The fourth step is to investigate how the model develops these representations over its training time, which is possible to do for models like Olmo, for which training checkpoints are released. This can enable us to compare the learning process, the order in which constructions are learned, to that of humans, giving us a further dimension of comparison to assess how human-like LLMs learn.

Using all these parameters of variation, we hope to be able to conduct more systematic research into why models fail to acquire some semantics of complex non-compositional constructions.

13.4.2 Abstraction and Learning of Constructions

This line of research is connected to that in Chapter 11 of investigating the learning mechanisms of LLMs in terms of analogy vs rules. We were able to do this for morphological derivation because we had different classes of adjectives (with respect to their -ity vs. -ness behaviour) with different levels of variability in the data, which gave us a controlled variable so that we could then make statements about the level of abstraction. These particular properties are not often found and we therefore have to find a new way to generalise the methodology.

The key idea is to artificially change the training data to study similar phenomena in a controlled setting. The BabyLM corpus (Warstadt et al., 2023) is a good prerequisite for this, as it was designed to be a cognitively plausible amount of training data, and its relatively small size allows us to train many different model variations on it. To study how constructions are learned, in the absence of "naturally controlled" phenomena like morphological derivation, we can identify all instances of a construction in the training data and change them systematically. For example, we might investigate if the construction is still learned if all of its instances are removed. We can also investigate the experimental results of Casenhiser and Goldberg (2005) that humans learn constructions more easily if its instances are not uniformly distributed but skewed.

Going even further than modifying the instances of a construction, we might also modify the construction itself, or create new ones. This can help us investigate the network of constructions that is hypothesised in CxG, but has not been worked out in detail in the literature (Hoffmann, 2017). In the spirit of Kallini et al. (2024), we can create constructions that we think are impossible and test if LLMs can acquire them. We can also create constructions that are plausible to exist based on all external factors, but don't, and attempt to make an LLM learn them.

[2] `https://huggingface.co/EleutherAI/gpt-j-6b`

[3] `https://www.kaggle.com/models/google/gemma`

Training many new models under different conditions will also enable us to observe their learning process at different stages, a more direct version of investigating the training checkpoints of large open models as described above.

13.4.3 New Methodologies for Probing Internal Representations

So far, investigations of the degree to which LLMs have learned constructions has focused either on generated output or on contextual embeddings at the last layer of the model. We have also described how we can investigate the models' learning process by applying the same test to the model at different checkpoints during training. Another dimension that we have yet to assess is that of the layers of the model: how are the meaning representations of large, non-compositional constructions built from smaller components, and how are they passed through the layers? A new line of work in interpretability seeks to provide methods for this, which we can adapt to investigate the learning of constructions. With logit lens (Nostalgebraist, 2020), the hidden states in the layer of any model can be decoded into a vocabulary distribution, enabling us to gain an insight into the processing state at that layer using natural language. Merullo et al. (2023) have demonstrated this for tasks such as naming the capital city of a country, or forming the past tense of a verb. They find distinct stages of processing in the model, the preparing of arguments and the surfacing of arguments. Similar methods could be employed to find the point in the layers where LLMs will apply the meaning of constructions.

13.4.4 Morphosyntax

All our work so far has been categorised into either Construction Grammar or Morphology. While this selection of areas of Linguistics might have seemed arbitrary, they are clearly connected, and we hope to make this more explicit in the future by conducting research into LLMs and *Morphosyntax*. As can be seen in the example of the Comparative Correlative in Chapters 5 and 6, constructions can have morphological constraints and components, such as requiring an adjective or adverb to be in the comparative. This means that LLMs' learning of them might be influenced by the properties of the tokenizer, which has been shown to influence performance on morphology (Hofmann et al., 2021) A possible direction for future research might be to find out if constructions with morphological constraints are more difficult to learn than those without, and how this is influenced by tokenization.

13.4.5 Constructions and Typology

William Croft's Morphosyntax, subtitled *Constructions of the World's Languages* (Croft, 2022) is a seminal work in typology, describing the constructions of all languages and how different strategies are used to express them. Croft uses the word *strategy* to signify the specific way in which a language expresses a meaning, for example, predication is expressed with a copula in some languages and without a copula in others. This grouping opens up fascinating avenues for CxG-based research of multilingual LLMs: do they form abstract representations at the level of constructions in Croft's sense, or at the level of strategies? How much do they acquire about the typological similarities between languages, and do they generalise better to low-resource languages when they have learned the parallel strategy for a high-resource one?

Chapter 14

Linguistic Contributions to NLP

Our second overarching research question was about the contributions that NLP and LLMs, in their current state of the art, can make to linguistics research and the creation of corpora. Specifically, we asked: "Given the current state of the art, how can NLP already contribute to Cognitive Linguistics?" **(Q2)**. We now summarise our contributions to this question for both Construction Grammar in Section 14.1 and Morphology in Section 14.2, summarise concurrent research in similar directions for both subfields, and then give an overview of exciting directions for future research.

14.1 Construction Grammar

Our contribution to CxG research with methods from modern NLP has been in the form of semi-automatic data annotation methods to create much-needed new corpora.

14.1.1 Our Results

In Section 1.3.2, we asked: "How can we use NLP to help annotate data for Construction Grammar?". In Chapter 8, we have introduced a new construction layer into Universal Dependencies. We have evaluated the feasibility of automatic annotation for ten languages and five constructions, and released the results, along with our proposal for annotation. As our work was done in conjunction with the UD core group, we hope that the new layer will be implemented swiftly and will make it possible for treebank maintainers to add their own rules and annotations, thereby harnessing an existing community to annotate data for CxG, where data has historically been sparse. Expanding beyond annotated treebanks, in Chapter 9, we have developed a novel hybrid annotation pipeline, which leveraged a dependency parser and ChatGPT. We have demonstrated that this greatly reduces human annotation cost and therefore enables the creation of larger corpora for CxG while preserving the quality control of manual annotation as the final step.

In this way, we have shown that although LLMs have not perfectly understood every construction, we can nonetheless use existing datasets, parsers, and LLMs along with human verification to aid in data creation.

14.1.2 Concurrent Research

Torrent et al. (2024) propose to use ChatGPT to help linguists investigate constructions. Notably, their main proposition is to ask the model to give examples of and explanations for a given construction. They investigate the diversity and accuracy of such examples. It is our position that it would be dangerous to treat these outputs as actual language data, and we therefore see their function more as a rubber duck for the linguist.

Most related to our work, Yan and Li (2023) have recently proposed a tool for the automatic extraction of constructions. The tool is highly interactive, and first provides a range of automatic annotations for a given corpus. It then presents statistically significant patterns to the user. Notably, it is not clear how well this technology will scale, as it is applied as an example to the relatively small BNC corpus (BNC Consortium, 2007).

14.2 Morphology

As a contribution of NLP methods to Morphology, we have studied unsupervised morphological induction using highly parallel corpora.

14.2.1 Our Results

In Section 1.4.2, we asked "How can we leverage parallel data for unsupervised morphology?" In Chapter 12, we have introduced the new task of unsupervised case marker extraction and built a baseline system for it, which leverages an automatic noun phrase annotation system and a highly parallel corpus. We have concluded that while our baseline has shown promise, this task remains challenging even with modern methods.

14.2.2 Concurrent Research

To the best of our knowledge, the only recent work related to ours is that of Kodner et al. (2023), who also evaluate LM-based models for morphological generalisation. In contrast to our work, they investigate the capabilities of purpose-built models, not general PLMs. This further highlights the research gap in evaluating PLMs for Morphology compared to other areas of Linguistics, such as Syntax or Semantics.

14.3 General Direction of the Field

The idea of using LLMs for linguistic annotation has seen increased attention since the release of ChatGPT. Gilardi et al. (2023) find that ChatGPT outperforms crowd-workers on tasks such as topic detection. Yu et al. (2023) and Savelka and Ashley (2023) evaluate the accuracy of GPT-3.5 and GPT-4 against human annotators, while Koptyra et al. (2023) annotate a corpus of data labelled for emotion by ChatGPT, but acknowledge its lower accuracy compared to a human-annotated version. Holter and Ell (2023) create a small gold standard for industry requirements by generating an initial parse tree with GPT-3 and then correcting it with a human annotator. Pangakis et al. (2023) investigate LLM annotation performance on 27 different tasks in two steps. First, annotators compile a codebook of annotation guidelines, which is then given to the LLM as help for annotation, and then the codebook is refined by the annotators in a second step. However, they find little to no improvement from the second step. Gray et al. (2023) make an LLM pre-generate labels for legal text analytics tasks which are then corrected by human annotators, but find that this does not speed up the annotation process.

14.4 Future Work

We do not see it as likely that automatic annotation methods will be adopted in Linguistics in the near future, as manual control from annotators is still key, and fully automated extraction or annotation pipelines will not be trusted in terms of accuracy or diversity. While we have shown that semi-automatic and hybrid methods are possible, they have all relied on finding researchers who are somewhat competent in using computational methods. The hybrid-human annotation pipeline that we have proposed in Chapter 9 requires computational infrastructure, a programmer to set up the dependency parsing, and competence in using the GPT API. The community annotation project for UD that we have described in Chapter 8 was made possible by a community of computational linguists who were able to operate the grew-match (Guillaume, 2021) website. These projects are promising, but are expected to reach mostly computational and corpus linguists. We therefore see it as an avenue for future research to turn these ideas into softwares or websites, for example, by making the dependency-parsed version of the reddit corpus available to search. This unfortunately raises infrastructure problems, as making corpora searchable through websites and putting the necessary software engineering work into making the tools accessible for linguists will probably not be possible for university research labs.

Zusammenfassung

In den letzten 100 Jahren gibt es in der Linguistik und der natürlichen Sprachverarbeitung (Natural Language Processing; NLP) eine Debatte über die Mechanismen, die den menschlichen sprachlichen Fähigkeiten zugrunde liegen, und die besten Methoden, sie rechnerisch zu repräsentieren. Sprachmodelle (Language Models; LMs) wurden sogar als Stellvertreter vorgeschlagen, die einfacher zu untersuchen sind als das menschliche Gehirn. Zunächst ist es jedoch notwendig zu bewerten, wie gut sie derzeit Sprache modellieren, und die Mechanismen zu untersuchen, durch die sie dies tun. Diese Arbeit schlägt vor, dies mit vielfältiger und neuartiger Methodologie aus der Linguistik zu tun, die es uns ermöglicht, seltener vorkommende und weniger zusammengesetzte Phänomene anzugehen, die die Modelle herausfordern können.

Um Methoden zur **Bewertung der sprachlichen Fähigkeiten von LMs** zu entwickeln, schlagen wir zunächst vor, ihre Fähigkeit zur Darstellung und zum Erlernen von Konstruktionen zu bewerten. Konstruktionen sind Form-Bedeutungs-Paarungen auf jeder Ebene der Granularität. Ein klassisches Beispiel für eine oft beschriebene Konstruktion ist der englische comparative correlative, d.h. *the x-er, the y-er*. Wir entwickeln neuartige Sondierungs- und Evaluierungsmethoden und zeigen, dass moderne LMs größtenteils die syntaktische Struktur von Konstruktionen erlernt haben. Selbst modernste große Sprachmodelle haben jedoch Schwierigkeiten mit der nicht-kompositionellen Bedeutung, die ihnen zugeordnet ist. Wir evaluieren auch die Fähigkeit von LMs zur morphologischen Generalisierung, dem Prozess, ein gelerntes Muster auf die Bildung neuer Wörter anzuwenden. Wir stellen fest, dass große Sprachmodelle zwar bemerkenswert menschenähnlich in ihrer Generalisierung auf neue Wörter sind, jedoch immer noch Fehler machen und sich auf andere Mechanismen als Menschen verlassen. Diese Ergebnisse zeigen, dass große Sprachmodelle zwar bemerkenswert nahe daran sind, ähnlich wie Menschen zu agieren, aber wir immer noch Bereiche finden können, in denen Verbesserungen notwendig sind.

Bei der Untersuchung dessen, **was die moderne NLP zur Linguistik beitragen kann**, gehen wir zunächst den Mangel annotierter Daten für die Konstruktionsgrammatik (Construction Grammar; CxG) an. Da es derzeit nicht möglich ist, Konstruktionen vollständig automatisch zu annotieren oder zu parsen, schlagen wir Strategien vor, bei denen Menschen in den Prozess einbezogen werden, um Linguisten bei der Erstellung von Korpora zu unterstützen. Wir zeigen die Ergebnisse eines Community-Projekts zur Einführung einer CxG-Ebene in die Universal Dependencies Treebanks vor. Darüber hinaus entwickeln wir eine hybride Annotationspipeline, die große Sprachmodelle verwendet, um den menschlichen Annotationsaufwand zu reduzieren und somit die kostengünstige Erstellung von Korpora für sehr seltene Phänomene zu ermöglichen. Schließlich zeigen wir, wie hochparallele Korpora für die unüberwachte Induktion morphologischer Struktur für Sprachen mit geringen Ressourcen genutzt werden können.

In Abschnitt 1.3.1 fragen wir: “Wie gut haben PLMs Konstruktionen gelernt?”. In unserer Arbeit zur Evaluierung von PLMs für Konstruktionen in den Kapiteln 4 bis 7 und Kapitel 9 haben wir gezeigt, dass selbst die größten Sprachmodelle immer noch Schwierigkeiten mit dem Verständnis der Bedeutung einiger Konstruktionen haben. Dies lag teilweise an der Gestaltung unserer Studien und bedeutet nicht, dass wir daraus schließen können, dass große Sprachmodelle Konstruktionen nicht verstehen. Wie wir im Kapitel 4 argumentiert haben, ist aus Sicht der Konstruktionsgrammatik (CxG) alles eine Konstruktion, und daher modellieren große Sprachmodelle bereits viele Konstruktionen gut. Die Konstruktionen, die wir im Detail untersucht haben (insbesondere den englischen Komparativkorrelativ, causal excess, licensed causal und caused-motion), wurden absichtlich ausgewählt, um die Grenzen von große Sprachmodelle zu testen, insbesondere hinsichtlich der Bedeutung von Konstruktionen. Wir haben festgestellt, dass die von uns vermuteten Grenzen derzeit existieren, da Modelle Schwierigkeiten hatten, die Bedeutung der Konstruktion auf neuartige Kontexte anzuwenden.

Im Abschnitt 1.4.1 fragen wir: “Haben PLMs eine menschenähnliche Fähigkeit zur morphologischen Generalisierung erworben?”Bei der Untersuchung der Grenzen von große Sprachmodelle hinsichtlich der Morphologie in den Kapiteln 10 und 11 haben wir ihre Fähigkeiten zur Bildung menschenähnlicher morphologischer Verallgemeinerungen für Pseudowörter untersucht. Bei der Prüfung der GPT-3.5 auf Flexionsmorphologie im Kapitel 10 stellten wir fest, dass sie selbst in Few-Shot-Einstellungen für Englisch und Deutsch Schwierigkeiten hatte, aber dies galt umso mehr für Türkisch und Tamil. Als wir das Experiment für GPT-4 wiederholten,[1] stellten wir fest, dass es für

[1] Diese Ergebnisse wurden nicht veröffentlicht und sind nur auf dem Poster verfügbar, das auf der EMNLP 2023 präsentiert wurde.

Englisch bei Zero-Shot- und Few-Shot-Einstellungen eine perfekte Trefferquote von 5 erreicht hatte, aber in der One-Shot-Einstellung schlechter abschnitt als GPT-3.5, aufgrund einer Übergneralisierung des gegebenen Beispiels. Bei der Bewertung von GPT-J und GPT-4 in Bezug auf derivationale Morphologie im Kapitel 11 stellten wir fest, dass beide nicht in der Lage waren, menschliche Beurteilungen zu Pseudowörtern zu approximieren. Wir untersuchten die Fehler genauer und stellten eine übermäßige Abhängigkeit von Token-Frequenz anstelle von Typ-Frequenz sowie eine suboptimale Tokenisierung fest.

Im Abschnitt 1.3.2 haben wir die Frage gestellt: "Wie können wir NLP nutzen, um Daten für die Konstruktionsgrammatik zu annotieren?". Im Kapitel 8 haben wir eine neue Konstruktionsebene in Universal Dependencies eingeführt. Wir haben die Machbarkeit automatischer Annotation für zehn Sprachen und fünf Konstruktionen evaluiert und die Ergebnisse zusammen mit unserem Vorschlag zur Annotation veröffentlicht. Da unsere Arbeit in Zusammenarbeit mit der UD-Kerngruppe durchgeführt wurde, hoffen wir, dass die neue Ebene schnell implementiert wird und es den Treebank-Verwaltern ermöglicht, ihre eigenen Regeln und Annotationen hinzuzufügen. Dadurch kann eine bestehende Community genutzt werden, um Daten für die Konstruktionsgrammatik zu annotieren, wo historisch gesehen wenig Daten vorhanden waren.

In Kapitel 9 haben wir über die annotierten Treebanks hinausgegriffen und eine neuartige hybride Annotationspipeline entwickelt, die einen Dependenzparser und ChatGPT nutzt. Wir haben gezeigt, dass dies die Kosten für die menschliche Annotation erheblich reduziert und somit die Erstellung größerer Korpora für die Konstruktionsgrammatik ermöglicht, während die Qualitätskontrolle der manuellen Annotation als abschließender Schritt erhalten bleibt.

Auf diese Weise haben wir gezeigt, dass, obwohl große Sprachmodelle nicht jede Konstruktion perfekt verstanden haben, wir dennoch bestehende Datensätze, Parser und große Sprachmodelle zusammen mit menschlicher Verifizierung nutzen können, um bei der Erstellung von Daten zu helfen.

Im Abschnitt 1.4.2 haben wir die Frage gestellt: "Wie können wir parallele Daten für die unüberwachte Morphologie nutzen?" Im Kapitel 12 haben wir die neue Aufgabe der unüberwachten Extraktion von Kasusmarkierungen vorgestellt und ein Baselinesystem dafür entwickelt, das ein automatisches Annotationssystem für Nominalphrasen und einen stark parallelen Korpus nutzt. Obwohl unser Baselinesystem vielversprechend ist, bleibt diese Aufgabe auch mit modernen Methoden weiterhin anspruchsvoll.

Abbildungsverzeichnis

Tabellenverzeichnis

Bibliography

Adam Albright. 2002. Islands of reliability for regular morphology: Evidence from Italian. *Language*, 78(4):684–709.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, 90(2):119–161.

Stephen R. Anderson. 1992. *A-Morphous Morphology*. Cambridge Studies in Linguistics. Cambridge University Press.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate. ArXiv:1409.0473 [cs.CL].

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190.

James Baker. 1975. The DRAGON system–an overview. *IEEE Transactions on Acoustics, speech, and signal Processing*, 23(1):24–29.

James K. Baker. 1990. *Stochastic modeling for automatic speech understanding*, page 297–307. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Timothy Baldwin, William Croft, Joakim Nivre, Agata Savary, Sara Stymne, and Ekaterina Vylomova. 2023. Universals of Linguistic Idiosyncrasy in Multilingual Computational Linguistics (Dagstuhl Seminar 23191). *Dagstuhl Reports*, 13(5):22–70.

Marco Baroni. 2022. On the proper role of linguistically oriented deep net analysis in linguistic theorising. In Shalom Lappin and Jean-Philippe Bernardy, editors, *Algebraic structures in natural language*, pages 1–16. CRC Press.

Jon Barwise. 1981. Some computational aspects of situation semantics. In *19th Annual Meeting of the Association for Computational Linguistics*, pages 109–111, Stanford, California, USA. Association for Computational Linguistics.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 830–839.

Yonatan Belinkov, Sebastian Gehrmann, and Ellie Pavlick. 2020. Interpretability and analysis in neural NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 1–5, Online. Association for Computational Linguistics.

Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi, editors. 2023. *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Singapore.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2006. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Yoshua Bengio, Jean-françcois Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Roux, and Marie Ouimet. 2003. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press.

Jean Berko. 1958. The child's learning of English morphology. *Word*, 14(2-3):150–177.

Katrien Beuls and Paul Van Eecke. 2025. Construction grammar and artificial intelligence. In Mirjam Fried and Kiki Nikiforidou, editors, *The Cambridge Handbook of Construction Grammar*, Cambridge Handbooks in Language and Linguistics, pages 543–571. Cambridge University Press, Cambridge.

Leonard Bloomfield. 1926. A set of postulates for the science of language. *International Journal of American Linguistics*, 15:195 – 202.

BNC Consortium. 2007. The British National Corpus, XML Edition. `http://hdl.handle.net/20.500.14106/2554`. Oxford Text Archive.

Franz Boas. 1889. On alternating sounds. *American Anthropologist*, 2(1):47–54.

Hans C. Boas. 2010. The syntax–lexicon continuum in construction grammar: A case study of English communication verbs. *Belgian Journal of Linguistics*, 24(1):54–82.

Hans C. Boas. 2011. Zum Abstraktionsgrad von Resultativkonstruktionen. In *Sprachliches Wissen zwischen Lexikon und Grammatik*, Jahrbuch / Institut für Deutsche Sprache, pages 37 – 69. De Gruyter.

Claire Bonial and Harish Tayyar Madabushi, editors. 2023. *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*. Association for Computational Linguistics, Washington, D.C.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Jeremy K. Boyd and Adele E. Goldberg. 2011. Learning what not to say: The role of statistical preemption and categorization in a-adjective production. *Language*, 87(1):55–83.

Cristiano Broccias. 2012. The syntax-lexicon continuum. In *The Oxford Handbook of the History of English*. Oxford University Press.

Patricia J. Brooks and Michael Tomasello. 1999. How children constrain their argument structure constructions. *Language*, 75(4):720–738.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. ArXiv:2005.14165 [cs.CL].

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. ArXiv:2303.12712 [cs.CL].

Joan L. Bybee and Dan I. Slobin. 1982. Rules and schemas in the development and use of the English past tense. *Language*, 58(2):265–289.

Basilio Calderone, Nabil Hathout, and Olivier Bonami. 2021. Not quite there yet: Combining analogical patterns and encoder-decoder networks for cognitively plausible inflection. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 274–282, Online. Association for Computational Linguistics.

Devin Casenhiser and Adele E. Goldberg. 2005. Fast mapping between a phrasal form and meaning. *Developmental Science*, 8(6):500–508.

N. Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

Noam Chomsky. 1957. *Syntactic Structures*. De Gruyter Mouton, Berlin, Boston.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT press.

Noam Chomsky. 1969. Some empirical assumptions in modern philosophy of language. In Ernest Nagel, Sidney Morgenbesser, Patrick Suppes, and Morton Gabriel White, editors, *Philosophy, Science, and Method*. St. Martin's Press.

Noam Chomsky. 1987. *Language and problems of knowledge: The Managua lectures*. MIT press.

Noam Chomsky. 1995. The minimalist program. *Cambridge, MA*.

Noam Chomsky. 2006. *Language and mind*. Cambridge University Press.

Noam Chomsky and Howard Lasnik. 1993. The theory of principles and parameters. In *Syntax: An International Handbook of Contemporary Research*, pages 506–569, Berlin • New York. De Gruyter Mouton.

Noam Chomsky, Ian Roberts, and Jeffrey Watumull. 2023. Noam Chomsky: The false promise of ChatGPT. *The New York Times*. March 8, 2023.

Gabriella Chronis, Kyle Mahowald, and Katrin Erk. 2023. A method for studying semantic construal in grammatical constructions with interpretable contextual embedding spaces. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 242–261, Toronto, Canada. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2007. Fast semantic extraction using a novel neural network architecture. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 560–567, Prague, Czech Republic. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? encoder-decoder neural networks as cognitive models of English past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877, Florence, Italy. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017a. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017b. Paradigm completion for derivational morphology. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 714–720, Copenhagen, Denmark. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

William Croft. 2001. *Radical construction grammar: Syntactic theory in typological perspective*. Oxford University Press, USA.

William Croft. 2022. *Morphosyntax: Constructions of the World's Languages*. Cambridge Textbooks in Linguistics. Cambridge University Press.

William Croft and D Alan Cruse. 2004. *Cognitive linguistics*. Cambridge University Press.

Peter W. Culicover and Ray Jackendoff. 1999. The view from the periphery: The English comparative correlative. *Linguistic Inquiry*, 30(4):543–571.

Robert Dale. 2021. GPT-3: What's it good for? *Natural Language Engineering*, 27(1):113–118.

Verna Dankers, Anna Langedijk, Kate McCurdy, Adina Williams, and Dieuwke Hupkes. 2021. Generalising to German plural noun classes, from the perspective of a recurrent neural network. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 94–108, Online. Association for Computational Linguistics.

Vittoria Dentella, Fritz Günther, and Evelina Leivada. 2023. Systematic testing of three language models reveals low language accuracy, absence of response stability, and a yes-response bias. *Proceedings of the National Academy of Sciences*, 120(51):e2309583120.

Daniel Deutsch, John Hewitt, and Dan Roth. 2018. A distributional and orthographic aggregation model for English derivational morphology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1938–1947, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan Dunn. 2017. Computational learning of construction grammars. *Language and cognition*, 9(2):254–292.

Daniel Edmiston. 2020. A systematic analysis of morphological content in BERT models for multiple languages. ArXiv:2004.03032 [cs.CL].

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Martin C. Emele and Remi Zajac. 1990. Typed unification grammars. In *COLING 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics*.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Charles J Fillmore. 1985. Syntactic intrusions and the notion of grammatical construction. In *Annual Meeting of the Berkeley Linguistics Society*, volume 11, pages 73–86.

Charles J Fillmore. 1986. Varieties of conditional sentences. In *Eastern States Conference on Linguistics*, volume 3, pages 163–182.

Charles J Fillmore. 1988. The mechanisms of "construction grammar". In *Annual Meeting of the Berkeley Linguistics Society*, volume 14, pages 35–55.

Charles J. Fillmore. 1989. Grammatical construction theory and the familiar dichotomies. In Rainer Dietrich and Carl F. Graumann, editors, *Language Processing in Social Context*, volume 54 of *North-Holland Linguistic Series: Linguistic Variations*, pages 17–38. Elsevier.

Charles J. Fillmore, Paul Kay, and Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64(3):501–538.

Charles J Fillmore, Russell Lee-Goldman, and Russell Rhodes. 2012. The framenet constructicon. In Hans Christian Boas and Ivan A Sag, editors, *Sign-based construction grammar*, pages 309–372. CSLI Publications Stanford.

Mirjam Fried and Kiki Nikiforidou, editors. 2025. *The Cambridge Handbook of Construction Grammar*. Cambridge Handbooks in Language and Linguistics. Cambridge University Press, Cambridge.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB dataset of diverse text for language modeling. ArXiv:2101.00027 [cs.CL].

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Adele E. Goldberg. 1992. *Argument Structure Constructions*. Ph.D. thesis, University of California, Berkeley.

Adele E Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press.

Adele E Goldberg. 2003. Constructions: A new theoretical approach to language. *Trends in cognitive sciences*, 7(5):219–224.

Adele E Goldberg. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press, USA.

Adele E. Goldberg. 2013. Constructionist approaches. In *The Oxford Handbook of Construction Grammar*. Oxford University Press.

Adele E. Goldberg. 2024. Usage-based constructionist approaches and large language models. *Constructions and Frames*, 16(2):220–254.

Adele E. Goldberg and Laura A. Michaelis. 2017. One among many: Anaphoric one and its relationship with numeral one. *Cognitive Science*, 41(S2):233–258.

Omer Goldman, David Guriel, and Reut Tsarfaty. 2022. (un)solving morphological inflection: Lemma overlap artificially inflates models' performance. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 864–870, Dublin, Ireland. Association for Computational Linguistics.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Morgan Gray, Jaromir Savelka, Wesley Oliver, and Kevin Ashley. 2023. Can GPT alleviate the burden of annotation? In *Legal Knowledge and Information Systems*, pages 157–166. IOS Press.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. ArXiv:2402.00838 [cs.CL].

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. ArXiv:2312.00752 [cs.LG].

Bruno Guillaume. 2021. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics.

Hubert Haider. 2023. Is Chat-GPT a Grammatically Competent Informant? Preprint, lingbuzz 007285.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Zellig S. Harris. 1965. *String analysis of sentence structure*. Papers on formal linguistics (The Hague). Mouton.

Martin Haspelmath and Andrea D. Sims. 2010. *Understanding Morphology*, 2 edition. Routledge, London.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. ArXiv:2006.03654 [cs.CL].

Thomas Herbst and Thomas Hoffmann. 2018. Construction grammar for students: A constructionist approach to syntactic analysis (CASA). *Yearbook of the German Cognitive Linguistics Association*, 6(1):197–218.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Hilpert. 2014. Collostructional analysis: Measuring associations between constructions and lexical elements. In Dylan Glynn and Justyna A. Robinson, editors, *Corpus Methods for Semantics: Quantitative Studies in Polysemy and Synonymy*, pages 391–404. John Benjamins Publishing Company. Accessed: 2025-10-28.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Charles F. Hockett. 1947. Problems of morphemic analysis. *Language*, 23(4):321–343.

Charles F. Hockett. 1954. Two models of grammatical description. *Word*, 10(2-3):210–234.

Thomas Hoffmann. 2017. Construction grammar as cognitive structuralism: the interaction of constructional networks and processing in the diachronic evolution of English comparative correlatives. *English Language and Linguistics*, 21(2):349–373.

Thomas Hoffmann and Graeme Trousdale. 2013. *The Oxford Handbook of Construction Grammar*. Oxford University Press.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020a. DagoBERT: Generating derivational morphology with a pretrained language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3848–3861, Online. Association for Computational Linguistics.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020b. Predicting the growth of morphological families from social and linguistic factors. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7273–7283, Online. Association for Computational Linguistics.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves BERT‘s interpretation of complex words. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3594–3608, Online. Association for Computational Linguistics.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics.

Valentin Hofmann, Hinrich Schütze, and Janet Pierrehumbert. 2020c. A graph auto-encoder model of derivational morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1127–1138, Online. Association for Computational Linguistics.

Ole Magnus Holter and Basil Ell. 2023. Human-machine collaborative annotation: A case study with GPT-3. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 193–206, Vienna, Austria. NOVA CLUNL, Portugal.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge.

Yoichi Ishibashi, Danushka Bollegala, Katsuhito Sudoh, and Satoshi Nakamura. 2023. Evaluating the robustness of discrete prompts. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2373–2384, Dubrovnik, Croatia. Association for Computational Linguistics.

Ray Jackendoff. 1977. *X syntax: A study of phrase structure*. MIT press.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5084, Singapore. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Laura A. Janda, Olga Lyashevskaya, Tore Nesset, Ekaterina Rakhilina, and Francis M. Tyers. 2018. A constructicon for russian. In *Constructicography*, pages 165–182. John Benjamins.

F. Jelinek, L. Bahl, and R. Mercer. 1975. Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, 21(3):250–256.

Fred Jelinek. 1990. Self-organized language modeling for speech recognition. In Alexander Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, San Mateo, CA.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. ArXiv:2310.06825 [cs.CL].

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. ArXiv:2401.04088 [cs.LG].

Daniel Saul Jurafsky. 1992. *An On-line Computational Model of Human Sentence Interpretation: A Theory of the Representation and Use of Linguistic Knowledge*. Ph.D. thesis, University of California, Berkeley.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Roni Katzir. 2023. Why large language models are poor theories of human linguistic cognition. a reply to Piantadosi (2023). Lingbuzz Preprint. `https://lingbuzz.net/lingbuzz/007190`.

Paul Kay. 1999. Grammatical constructions and linguistic generalizations: The what's X doing Y? construction. *Language*, 75(1):1–33.

Paul Kay and Laura A. Michaelis. 2019. Constructional meaning and compositionality. In Claudia Maienborn, Klaus Heusinger, and Paul Portner, editors, *Semantics - Interfaces*, pages 293–324. De Gruyter Mouton, Berlin, Boston.

Paul Kay and Ivan A Sag. 2012. Cleaning up the big mess: Discontinuous dependencies and complex determiners. In Hans C. Boas and Ivan A. Sag, editors, *Sign-based construction grammar*, chapter 5, pages 229–256. CSLI Publications/Center for the Study of Language and Information.

Najoung Kim and Paul Smolensky. 2021. Testing for grammatical category abstraction in neural language models. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 467–470, Online. Association for Computational Linguistics.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Jordan Kodner, Salam Khalifa, and Sarah Ruth Brogden Payne. 2023. Exploring linguistic probes for morphological generalization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8933–8941, Singapore. Association for Computational Linguistics.

Bartłomiej Koptyra, Anh Ngo, Łukasz Radlinski, and Jan Kocon. 2023. CLARIN-emo: Training emotion recognition models using human annotation and chatGPT. In *Computational Science – ICCS 2023*, pages 365–379, Cham. Springer Nature Switzerland.

George Lakoff. 1987. *Women, fire, and dangerous things: What categories reveal about the mind*. University of Chicago press.

Ronald W. Langacker. 1986. An introduction to cognitive grammar. *Cognitive Science*, 10(1):1–40.

Adrieli Laviola, Ludmila Lage, Natália Marção, Tatiane Tavares, Vânia Almeida, Ely Matos, and Tiago Torrent. 2017. The Brazilian Portuguese constructicon: Modeling constructional inheritance, frame evocation and constraints in FrameNet Brasil. In *2017 AAAI Spring Symposium Series*.

Bai Li, Zining Zhu, Guillaume Thomas, Frank Rudzicz, and Yang Xu. 2022. Neural reality of argument structure constructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7410–7423, Dublin, Ireland. Association for Computational Linguistics.

Ling Liu and Lingshuang Jack Mao. 2016. Morphological reinflection with conditional random fields and unsupervised features. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 36–40, Berlin, Germany. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. ArXiv:1907.11692 [cs.CL].

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago Timponi Torrent, editors. 2018. *Constructicography: Constructicon development across languages*. John Benjamins Publishing Company.

Harish Tayyar Madabushi, Laurence Romain, Petar Milin, and Dagmar Divjak. 2025. Construction grammar and language models. In Mirjam Fried and Kiki Nikiforidou, editors, *The Cambridge Handbook of Construction Grammar*, Cambridge Handbooks in Language and Linguistics, pages 572–595. Cambridge University Press, Cambridge.

Kyle Mahowald. 2023. A discerning several thousand judgments: GPT-3 rates the article + adjective + numeral + noun construction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 265–273, Dubrovnik, Croatia. Association for Computational Linguistics.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Gary F. Marcus, Ursula Brinkmann, Harald Clahsen, Richard Wiese, and Steven Pinker. 1995. German inflection: The exception that proves the rule. *Cognitive Psychology*, 29(3):189–256.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland. European Language Resources Association (ELRA).

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK. Coling 2008 Organizing Committee.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

W. A. Martin. 1980. Parsing. In *18th Annual Meeting of the Association for Computational Linguistics*, pages 91–93, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Peter H. Matthews. 1991. *Morphology*, 2 edition. Cambridge Textbooks in Linguistics. Cambridge University Press.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

Warren S McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133.

Kate McCurdy, Sharon Goldwater, and Adam Lopez. 2020. Inflecting when there‘s no majority: Limitations of encoder-decoder neural networks as cognitive models for German plurals. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1745–1756, Online. Association for Computational Linguistics.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. A mechanism for solving relational tasks in transformer language models. ArXiv:2305.16130 [cs.CL].

Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. ArXiv:1301.3781 [cs.CL].

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan ernocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531.

Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.

George A. Miller and Noam Chomsky. 1963. Finitary models of language users. In D. Luce, editor, *Handbook of Mathematical Psychology*, pages 2–419. John Wiley & Sons.

Kanishka Misra and Najoung Kim. 2023. Abstraction via exemplars? a representational case study on lexical category inference in BERT. ArXiv:2312.03708 [cs.CL].

R. K. Moore. 2005. Results from a survey of attendees at ASRU 1997 and 2003. In *Proceedings of Interspeech 2005*, pages 117–120.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022. Probing for labeled dependency trees. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7711–7726, Dublin, Ireland. Association for Computational Linguistics.

Gunter Neumann and Wolfgang Finkler. 1990. A head-driven approach to incremental and parallel generation of syntactic structures. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC‘16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

The Nostalgebraist. 2020. Interpreting GPT: The logit lens. Blogpost accessed: 2024-03-05.

K. H. Ohara, S. Fujii, T. Ohori, R. Suzuki, H. Saito, and S. Ishizaki. 2004. The Japanese FrameNet project: An introduction. In *Proceedings of LREC-04 Satellite Workshop "Building Lexical Resources from Semantically Annotated Corpora"(LREC 2004)*, pages 9–11.

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt`. Accessed: June 22, 2023.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 technical report. ArXiv:2303.08774 [cs.CL].

Martha Palmer. 1981. A case for rule-driven semantic processing. In *19th Annual Meeting of the Association for Computational Linguistics*, pages 125–131, Stanford, California, USA. Association for Computational Linguistics.

Nicholas Pangakis, Samuel Wolken, and Neil Fasching. 2023. Automated annotation with generative AI requires validation. ArXiv:2306.00176 [cs.CL].

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Steven Pinker. 1999. *Words and Rules: The Ingredients of Language*. Basic Books, New York.

Steven Pinker and Alan Prince. 1988. On Language and Connectionism: Analysis of a Parallel Distributed Processing Model of Language Acquisition. *Cognition*, 28(1-2):73–193.

Christopher Potts. 2024. Characterizing English preposing in PP constructions. *Journal of Linguistics*, page 1–39.

Sandeep Prasada and Steven Pinker. 1993. Generalisation of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8(1):1–56.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Unpublished preprint (OpenAI).

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. Unpublished preprint (OpenAI).

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Rudolf Rosa, Jan Mašek, David Marecek, Martin Popel, Daniel Zeman, and Zdenek Žabokrtský. 2014. HamleDT 2.0: Thirty dependency treebanks stanfordized. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2334–2341, Reykjavik, Iceland. European Language Resources Association (ELRA).

Eleanor H Rosch. 1973. Natural categories. *Cognitive psychology*, 4(3):328–350.

David E. Rumelhart and James L. McClelland. 1986. On learning the past tenses of English verbs. In David E. Rumelhart, James L. McClelland, and PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 216–271. MIT Press, Cambridge, MA.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Edward Sapir. 1929. The status of linguistics as a science. *Language*, 5:207–214.

Jaromir Savelka and Kevin D Ashley. 2023. The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts. *Frontiers in Artificial Intelligence*, 6.

Hans-Jörg Schmid. 2007. Non-compositionality and emergent meaning of lexico-grammatical chunks: A corpus study of noun phrases with sentential complements as constructions. *Zeitschrift für Anglistik und Amerikanistik*, 55(3):313–340.

Hans-Jörg Schmid. 2020. *The Dynamics of the Linguistic System: Usage, Conventionalization, and Entrenchment*. Oxford University Press.

Hans-Jörg Schmid and Helmut Küchenhoff. 2013. Collostructional analysis and other ways of measuring lexicogrammatical attraction: Theoretical premises, practical problems and cognitive underpinnings. *Cognitive Linguistics*, 24(3):531–577.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Hinrich Schütze. 1992. Dimensions of meaning. In *SC Conference*, pages 787–796, Los Alamitos, CA, USA. IEEE Computer Society.

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Seventh Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Paul Smolensky, Richard McCoy, Roland Fernandez, Matthew Goldrick, and Jianfeng Gao. 2022. Neurocompositional computing: From the central paradox of cognition to a new generation of ai systems. *AI Magazine*, 43(3):308–322.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Norman K. Sondheimer, editor. 1979. *17th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, La Jolla, California, USA.

Norman K. Sondheimer, editor. 1980. *18th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Morris Swadesh. 1950. Salish internal relationships. *International Journal of American Linguistics*, 16(4):157–167.

Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets construction grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, Hyun-Jeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozi ska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lu i , Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Ça lar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Raki evi , Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz K pa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei LouisChen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan

Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybi ski, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Pluci ska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini: A family of highly capable multimodal models. ArXiv:2312.11805 [cs.CL].

Tiago Timponi Torrent, Thomas Hoffmann, Arthur Lorenzi Almeida, and Mark Turner. 2024. *Copilots for Linguists: AI, Constructions, and Frames.* Elements in Construction Grammar. Cambridge University Press.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. ArXiv:2302.13971 [cs.CL].

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. ArXiv:2307.09288 [cs.CL].

Trieu H. Trinh and Quoc V. Le. 2019. A simple method for commonsense reasoning. ArXiv:1806.02847 [cs.CL].

Yu-Hsiang Tseng, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. CxLM: A construction and context-aware language model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6361–6369, Marseille, France. European Language Resources Association.

Friedrich Ungerer and Hans-Jörg Schmid. 2006. *An Introduction to Cognitive Linguistics*, 2 edition. Routledge, London.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tim Veenboer and Jelke Bloem. 2023. Using collostructional analysis to evaluate BERT‘s representation of linguistic constructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12937–12951, Toronto, Canada. Association for Computational Linguistics.

Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin, and Trevor Cohn. 2017. Context-aware prediction of derivational word-forms. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 118–124, Valencia, Spain. Association for Computational Linguistics.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`. GitHub repository.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adam Wiemerslage, Shiran Dudy, and Katharina Kann. 2022. A comprehensive comparison of neural networks as cognitive models of inflection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1933–1945, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Adam Wiemerslage, Changbing Yang, Garrett Nicolai, Miikka Silfverberg, and Katharina Kann. 2023. An investigation of noise in morphological inflection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3351–3365, Toronto, Canada. Association for Computational Linguistics.

Colin Wilson and Jane S.Y. Li. 2021. Were we there already? Applying minimal generalization to the SIGMORPHON-UniMorph shared task on cognitively plausible morphological inflection. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 283–291, Online. Association for Computational Linguistics.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Zhilin Gong, Ming Cai, and Tianxiang Wang. 2023. Enhancing language representation with constructional information for natural language understanding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4685–4705, Toronto, Canada. Association for Computational Linguistics.

Hengbin Yan and Yinghui Li. 2023. Constraction: a tool for the automatic extraction and interactive exploration of linguistic constructions. *Linguistics Vanguard*, 9(1):215–227.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 207–216, Hong Kong. Association for Computational Linguistics.

Danni Yu, Luyang Li, Hang Su, and Matteo Fuoli. 2023. Assessing the potential of LLM-assisted annotation for corpus-based pragmatics and discourse analysis: The case of apologies. *International Journal of Corpus Linguistics*.

Anthony Zador, Sean Escola, Blake Richards, Bence Ölveczky, Yoshua Bengio, Kwabena Boahen, Matthew Botvinick, Dmitri Chklovskii, Anne Churchland, Claudia Clopath, et al. 2023. Catalyzing next-generation artificial intelligence through NeuroAI. *Nature communications*, 14(1):1597.

WD Zhan. 2017. On theoretical issues in building a knowledge database of Chinese constructions. *Journal of Chinese Information Processing*, 31(1):230–238.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Alexander Ziem, Johanna Flick, and Phillip Sandkühler. 2019. The German constructicon project: Framework, methodology, resources. *Lexicographica*, 35(2019):15–40.

Alexander Ziem and Alexander Lasch. 2013. *Konstruktionsgrammatik: Konzepte und Grundlagen gebrauchsbasierter Ansätze*. De Gruyter, Berlin, Boston.