Exploring the Frontiers of Word Understanding and Language Model Evaluation in NLP

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

vorgelegt von
Lütfi Kerem Şenel
aus Ankara

München, den 4 June 2024

Erstgutachter: *Prof. Dr. Hinrich Schütze*
Zweitgutachter: *Prof. Dr. Goran Glavaš*
Drittgutachter: *Prof. Dr. Benjamin Roth*

Tag der Einreichung: 4 June 2024
Tag der mündlichen Prüfung: 28 Februar 2025

Eidesstattliche Versicherung

(siehe Promotionsordnung vom 12.07.11, §8, Abs. 2 Pkt. 5.)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig ohne unerlaubte Beihilfe angefertigt ist.

München, den 4 June 2024

Lütfi Kerem Şenel

Abstract

The field of natural language processing (NLP) has progressed dramatically with the rise of deep learning, yet many challenges in learning high-quality semantic representations remain. This thesis addresses these challenges through a series of studies focusing on both monolingual and multilingual contexts.

First, to improve transfer learning for low-resource languages, we develop methodologies utilizing Turkish as a high-resource proxy for related Turkic languages. By creating and evaluating on the new Kardes-NLU benchmark, we demonstrate substantial performance gains when Turkish is integrated at both intermediate training and fine-tuning stages, underscoring the value of leveraging linguistic relatives in cross-lingual transfer.

Second, we explore the underutilized potential of multiparallel corpora for enhancing word alignment. By constructing word alignment graphs from over 80 language pairs and applying advanced graph algorithms, including graph neural networks, we significantly improve alignment accuracy, showcasing the benefits of community detection techniques in multilingual settings.

Third, we introduce BiImp, a novel method to enhance the interpretability of word embeddings by aligning dimensions with semantic concepts derived from lexical databases like WordNet and Roget's Thesaurus. This approach enables the creation of interpretable embeddings that maintain high performance and reduces biases, such as gender bias.

Finally, we focus on developing robust evaluation measures for language models. We introduce WDLMPro and CoDA21, two challenging benchmarks that assess a model's ability to match words with definitions and align context with definitions without prior word knowledge, respectively. These benchmarks reveal significant performance gaps between models and human understanding, highlighting critical areas for improvement in language comprehension.

This thesis significantly contributes to the field by enhancing the quality of semantic representations in NLP, improving transfer strategies for low-resource languages, advancing word alignment methods, increasing interpretability of embeddings, and developing more nuanced evaluation benchmarks.

Zusammenfassung

Das Gebiet der Verarbeitung natürlicher Sprache (NLP) hat mit dem Aufstieg des Deep Learning dramatische Fortschritte gemacht, doch viele Herausforderungen bei der Erzeugung qualitativ hochwertiger semantischer Repräsentationen bleiben bestehen. Diese Dissertation adressiert diese Herausforderungen durch eine Reihe von Studien, die sich sowohl auf monolinguale als auch auf mehrsprachige Kontexte konzentrieren.

Erstens entwickeln wir, um das Transferlernen für ressourcenarme Sprachen zu verbessern, Methoden, die Türkisch als ressourcenreiches Proxy für verwandte türkische Sprachen nutzen. Durch die Erstellung und Evaluierung anhand des neuen Kardes-NLU-Benchmarks zeigen wir erhebliche Leistungssteigerungen, wenn Türkisch sowohl in den Zwischen- als auch in den Feinabstimmungsphasen integriert wird, was den Wert der Nutzung sprachlicher Verwandtschaften im cross-lingualen Transfer unterstreicht.

Zweitens untersuchen wir das ungenutzte Potenzial multiparalleler Korpora zur Verbesserung der Wortausrichtung. Durch die Erstellung von Wortausrichtungsgraphen aus über 80 Sprachpaaren und die Anwendung fortgeschrittener Graphenalgorithmen, einschließlich graphneurale Netzwerke, verbessern wir die Ausrichtungsgenauigkeit erheblich und zeigen die Vorteile der Community-Detection-Techniken in mehrsprachigen Umgebungen.

Drittens führen wir BiImp ein, eine neuartige Methode zur Verbesserung der Interpretierbarkeit von Wort-Embeddings, indem Dimensionen mit semantischen Konzepten aus lexikalischen Datenbanken wie WordNet und Roget’s Thesaurus abgeglichen werden. Dieser Ansatz ermöglicht die Erstellung interpretierbarer Embeddings, die eine hohe Leistung beibehalten und Vorurteile wie Geschlechtervorurteile reduzieren.

Schließlich konzentrieren wir uns auf die Entwicklung robuster Evaluierungsmaßnahmen für Sprachmodelle. Wir stellen WDLMPro und CoDA21 vor, zwei herausfordernde Benchmarks, die die Fähigkeit eines Modells bewerten, Wörter mit Definitionen abzugleichen und den Kontext mit Definitionen ohne vorherige Wortkenntnis in Übereinstimmung zu bringen. Diese Benchmarks zeigen signifikante Leistungslücken zwischen Modellen und menschlichem Verständnis auf und heben kritische Verbesserungsbereiche im Sprachverständnis hervor.

Diese Dissertation trägt erheblich zum Feld bei, indem sie die Qualität semantischer Repräsentationen in der NLP verbessert, Transferstrategien für ressourcenarme Sprachen weiterentwickelt, Methoden zur Wortausrichtung vorantreibt, die Interpretierbarkeit von Embeddings erhöht und nuanciertere Evaluationsbenchmarks entwickelt.

Publications and Declaration of Co-Authorship

Chapter 2 corresponds to the following publication:

> **Lütfi Kerem Şenel***, Benedikt Ebing*, Konul Baghirova, Hinrich Schuetze, Goran Glavaš *Kardeş-NLU: Transfer to Low-Resource Languages with Big Brother's Help – A Benchmark and Evaluation for Turkic Languages*. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2024. *: equal contribution

Goran Glavaš conceived of the original research contributions and performed implementation and evaluation. Konul Baghirova worked on the dataset creation. I wrote the initial draft of the paper and did most of the subsequent corrections together with Benedikt Ebing. I regularly discussed this work with my coauthors and my advisor, who assisted in improving the draft.

Chapter 3 corresponds to the following publication:

> Ayyoob Imani, Masoud Jalili Sabet, **Lütfi Kerem Şenel**, Philipp Dufter, François Yvon, and Hinrich Schütze. *Graph Algorithms for Multiparallel Word Alignment.* In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2021.

Ayyoob Imani and Masoud Jalili Sabet conceived of the original research contributions. I performed the experiments for the analysis of the model (Section 5.3). All authors wrote the initial draft of the article and did the subsequent corrections. All authors regularly discussed this work with each other and improved the draft.

Chapter 4 corresponds to the following publication:

> Ayyoob Imani, **Lütfi Kerem Şenel**, Masoud Jalili Sabet, François Yvon, and Hinrich Schuetze. *Graph Neural Networks for Multiparallel*

Word Alignment. In Findings of the Association for Computational Linguistics: ACL 2022.

Ayyoob Imani conceived of the original research contributions. I regularly discussed this work with my coauthors and helped Ayyoob Imani with the evaluation of the model. I wrote the initial draft of the paper. My coauthors assisted me in improving the draft.

Chapter 5 corresponds to the following publication:

> **Lütfi Kerem Şenel**, Furkan Şahinuç, Veysel Yücesoy, Hinrich Schütze, Tolga Çukur, and Aykut Koç. *Learning interpretable word embeddings via bidirectional alignment of dimensions with semantic concepts.* Information Processing and Management, 59, 3 (May 2022).

This publication is based on my work on imparting interpretability to word embeddings. I performed most of the implementation and the evaluation for this work. Along with my coauthors, I co-wrote the published paper.

Chapter 6 corresponds to the following publication:

> **Lütfi Kerem Şenel** and Hinrich Schütze. *Does She Wink or Does She Nod? A Challenging Benchmark for Evaluating Word Understanding of Language Models.* In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021.

I regularly discussed this work with my advisor, but I conceived of the original research contributions and performed implementation and evaluation. I wrote the initial draft of the paper and did most of the subsequent corrections. My advisor assisted in improving the draft.

Chapter 7 corresponds to the following publication:

> **Lütfi Kerem Şenel**, Timo Schick, and Hinrich Schuetze. *CoDA21: Evaluating Language Understanding Capabilities of NLP Models With Context-Definition Alignment.* In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Main Volume, ACL 2022.

I conceived of the original research contributions and performed implementation and evaluation. I wrote the initial draft of the paper and did most of the subsequent corrections. I regularly discussed this work with my coauthors and my advisor, who assisted in improving the draft.

Declaration on Writing Aids

I used GPT-4-Turbo (also referred to as “GPT” and the “AI tool” below) as an aid to write this thesis. Throughout the thesis, excluding the papers, GPT-4-Turbo was used as an advanced spelling and grammar checker: text was passed to GPT to correct spelling and grammar, to improve choice of words, to restructure sentences and to improve clarity and readability.

For writing parts of the introduction chapter, my use was particularly extensive as discussed below.

Motivation Section. In the motivation section, I used GPT-4-Turbo to revise and polish my initial draft. The AI tool corrected grammatical errors and rephrased poorly constructed sentences. This was supplemented by AI-facilitated brainstorming sessions, where the tool helped in generating ideas on what topics and arguments should be included. This process was iterative, involving a mix of AI-generated suggestions and personal content creation, which I then refined to align with my outline and my goals for this section.

Background Section. For the background section, I provided GPT-4-Turbo with general instructions, bullet points, initial snippets of content and the main insights I wanted the section to focus on and then went through multiple iterations of enhancements until the resulting text was a satisfactory rough draft. I then carefully rewrote and edited it to finalize the section.

Mathematical Equations. For formulas that are standard and widely used in deep learning and NLP, I utilized GPT-4-Turbo for generating LaTeX source code. To this end, I defined the notation manually and then prompted GPT to draft the equations. Each equation was subsequently manually verified and (if needed) edited for accuracy and relevance to the thesis content, ensuring mathematical integrity and appropriateness.

Contents

Chapter 1

Introduction

1.1 Motivation

Language is arguably humanity's most significant innovation, an essential tool that has shaped civilizations, cultures, and personal experiences throughout history. In the modern era, the digital revolution has amplified the importance of language, making it the primary channel through which we share information, conduct transactions, and facilitate interactions around the globe. As technology continues to advance and become ever more integrated into our daily lives, the ability to effectively process and understand human language using computational methods has emerged as a critical challenge and opportunity. This challenge lies at the heart of natural language processing (NLP), a rapidly evolving field situated at the crossroads of computer science, linguistics, and artificial intelligence (Jurafsky and Martin, 2000; Bengio et al., 2003).

Natural language processing endeavors to empower machines with the capacity to parse, comprehend, and generate human language, mirroring the nuance and richness of human communication. The motivation driving NLP research is not only technological but deeply rooted in the desire to bridge the communicative gap between humans and machines, facilitating interactions that are as seamless and intuitive as possible. From powering search engines and virtual assistants to enabling sophisticated machine translation and sentiment analysis, NLP technologies are ubiquitous in today's digital landscape, transforming how we access and interact with information and each other.

The journey from the field's inception to its current state is one of relentless innovation, punctuated by shifts from rule-based methodologies to statistical models, and more recently, to the advent of deep learning paradigms (Manning and Schütze, 1999; Goldberg, 2017; Goodfellow et al., 2016). These shifts reflect both advancements in computational capacities and a growing understanding of language's inherent complexities. Notably, the transition towards data-driven techniques, especially through neural networks and the exploitation of large textual datasets, has significantly reshaped NLP. With motivations rooted in the distributional hypothesis (Harris, 1954; Firth, 1957), modern approaches excel at learning powerful, dense semantic representations for linguistic units, such as words, sentences or documents, directly from text, circumventing the need for exhaustive manual rule crafting. This ability to autonomously discern and utilize intricate patterns within language has catapulted NLP forward, enabling a broad spectrum of applications to achieve remarkable levels of performance and flexibility.

Despite the field's impressive progress, and remarkable performance of learned language representations on various tasks, numerous challenges remain in learning high quality representations for NLP applications. The lifecycle of a typical neural network based NLP model involves several stages, from data collection and preprocessing to model training and evaluation. At the core of learning a model with high quality representations is the dependency on large, high quality textual corpora which are predominantly available for high-resource languages like English. This poses a significant limitation for low-resource languages, which are substantial in number and critical for global inclusivity. An essential aspect of ongoing research involves developing methods to transfer knowledge from high-resource to low-resource languages effectively (Lauscher et al., 2020; Parović et al., 2022; Schmidt et al., 2022). However, the effectiveness of these transfer strategies varies due to the linguistic and typological distances between languages, necessitating more sophisticated adaptation techniques.

Instead of transferring knowledge from a model that is trained on a high-resource language, another approach is to leverage the information from multiple languages simultaneously to learn *multilingual representations* that can capture the shared semantic and syntactic properties of the languages. One valuable type of resource for learning high quality multilingual representations is parallel corpora, which contain translations of the same text in different languages. These corpora provide valuable information about the relationships between words in different languages, enabling models to learn cross-lingual representations that capture the shared semantic and syntactic properties of the languages which is crucial for many multilingual NLP tasks such as word alignment and machine translation. Although parallel corpora have been used extensively in NLP research, it is often overlooked that most parallel corpora are indeed multi-parallel, containing translations in more than two languages, exploitation of which can lead to even better quality word alignments and cross-lingual representations.

Another important issue with modern NLP models is interpretability. The rise of deep learning in NLP has introduced models that, while delivering state-of-the-art results, often act as "black boxes." These models provide limited insight into their decision-making processes, as these models become more sophisticated and capable, ensuring their safety and controllability becomes ever more crucial, especially in sensitive or high-stakes applications. Addressing this, efforts in making these models more interpretable and transparent continue to be a priority to ensure their reliability and ethical use (Sydorova et al., 2019; Poerner et al., 2018).

Once models are trained, evaluating their performance is one of the most critical aspects of NLP research. This is crucial to understand the extent of their capabilities and to identify their shortcomings in order to develop novel techniques to address them. Many evaluation benchmarks have been proposed to assess the performance of NLP models, but they often fail to capture the full range of linguistic phenomena that models must handle in real-world applications. As a result, the models that seemingly achieve superhuman performance on these benchmarks often fail to generalize to more challenging tasks or real-world scenarios. This discrepancy highlights the need for more comprehensive evaluation resources that can provide a more nuanced understanding of the capabilities and limitations of NLP models.

1.1.1 Research Questions

In order to have reliable and efficient NLP systems, we need to ensure the high quality of the learned semantic representations which is a multi-faceted problem. In this thesis, we aim to address the following research questions, categorized into 4 groups, in order to improve the quality of learned semantic representations and their applications in NLP:

(i) **Transferring to Low-Resource:** How can we improve the transfer to low resource languages? Can we develop more effective strategies to leverage high-resource languages for better transfer to low-resource languages using their high resource relatives?

(ii) **Multiparallelity:** How can we use multi-parallel corpora for multilingual tasks? In particular, is it possible to improve word alignment by using synergies in multi-parallel corpora?

(iii) **Interpretability:** How can we learn interpretable word representations? Is it possible to learn highly interpretable and yet powerful word representations?

(iv) **Evaluation:** How can we effectively evaluate language understanding capabilities of language models? Can we design more challenging evaluation benchmarks that can provide a more nuanced understanding of the models' capabilities and limitations?

1.1.2 Approach and Contributions

In this work we tackle some of the important challenges in learning high quality semantic representations for NLP applications (as identified in Section 1.1.1) by proposing novel methods and evaluation benchmarks in both monolingual and multilingual settings.

Within the scope of obtaining high quality representations for low resource languages, to address (i) we investigate novel transfer strategies. Specifically, we focus on 5 low-resource Turkic languages: Azerbaijani, Kazakh, Kyrgyz, Uyghur and Uzbek, and experiment with different transfer strategies by leveraging Turkish as the high resource relative of the low resource languages. Starting with the popular multilingual language model XLM-R (Conneau et al., 2020), we investigate the effectiveness of incorporating the high resource language Turkish into the (i) continual pretraining and (ii) fine-tuning stages of the model for zero shot cross-lingual transfer to the low resource languages. Since these low resource languages lack high quality evaluation resources, we also create new evaluation benchmarks for these languages by translating popular English benchmarks to the target languages using machine translation followed by manual curation. We then evaluate the performance of our transfer strategies on these new evaluation benchmarks. Our findings show that a related language with higher resources can facilitate better transfer to low resource languages.

Focusing our attention to resources for learning high quality multilingual representations, to address (ii), we propose novel methods to exploit multi-parallel corpora for improving word alignment quality. We start by obtaining bilingual alignments between pairs of over 80 different languages using the state-of-the-art word alignment methods and taking bible as the parallel corpus. Then, using the bilingual alignments between each pair of languages, we construct a graph where nodes represent words in different languages and edges represent the alignment between the words. By applying the graph algorithms to this graph, we show that we can improve the quality of word alignments between languages by leveraging the information from multiple languages to find missing alignment links. We further extend this work by proposing a novel method that uses community features and graph neural networks that can make use of the semantics of the words to improve the quality of word alignments.

To tackle the challenge of resolving the black box nature of modern NLP models and have better insights about their nature, in (iii) we focus on interpretability, specifically interpretability of word representations known as *word embeddings*. We propose a novel method for learning interpretable word embeddings by bidirectionally aligning the embedding dimensions with semantic concepts. we make use of existing lexical resources, namely WordNet (Miller, 1994) and Roget's Thesaurus (Roget, 2008), to extract any desired number of interpretable and distinct semantic concepts as word lists. We, then, using our proposed modification, called BiImp, to the training objective of popular word embedding methods, word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014), bidirectionally align the dimensions of word embeddings with semantic concepts during the learning of the word vectors from corpora. Through comprehensive evaluations, we show that the proposed method can learn interpretable word embeddings that are aligned with the desired semantic concepts, while maintaining competitive performance. We also show that encoding of a concept like gender into one particular dimension of the word embeddings can help facilitate better removal of gender bias from the embeddings.

We, finally, shift our focus from enhancing NLP models in various ways to evaluating their capabilities to tackle (iv). By utilizing the hyponym-hypernym relationships encoded between the synsets in WordNet, we introduce two new challenging evaluation benchmarks, Word Definition Language Model Probing (WDLMPro) and Context-Definition Alignment (CoDA21). WDLaMPro assesses the model's understanding of meaning of words by testing whether it can match the words and their definitions in a challenging setting. CoDA21 takes a step further by evaluating the model's language understanding capabilities by testing whether it can align the context of a word with its definition without knowing the word itself. We show that the proposed benchmarks can provide a more nuanced understanding of the models' capabilities and limitations compared to existing benchmarks.

1.2 Outline of the Thesis

In this chapter we motivate and introduce our work as well as provide the necessary background information and foundational notation for the models used in the subsequent chapters. In Chapter 2, we investigate novel transfer strategies to low-resource languages by leveraging a related high resource language and we also introduce a new evaluation benchmarks those low-resource languages. In Chapter 3 and Chapter 4, we propose novel methods for exploiting multi-parallel corpora to improve word alignment quality. In Chapter 5, we introduce a novel method for learning interpretable word embeddings by bidirectionally aligning the embedding dimensions with semantic concepts. Finally, in Chapter 6 and Chapter 7, we introduce new evaluation benchmarks for assessing the word and context understanding capabilities of language models.

1.3 Background

1.3.1 Mathematical Notation

Scalar quantities are denoted by lowercase italic characters, such as $t \in \mathbb{R}$, vectors are depicted by bold lowercase letters, for example, $\mathbf{v} \in \mathbb{R}^d$, and matrices are illustrated with bold uppercase letters, such as $\mathbf{W} \in \mathbb{R}^{d \times n}$. The i-th component of a vector $\mathbf{v}$ is referred to as v_i. The i-th row and j-th column of a matrix $\mathbf{W}$ are denoted by $\mathbf{W}_{i,\cdot}$ and $\mathbf{W}_{\cdot,j}$ respectively. The element in the i-th row and j-th column of a matrix $\mathbf{W}$ is represented by $\mathbf{W}_{i,j}$. The transpose of a vector and a matrix are represented by $\mathbf{x}^T$ and $\mathbf{X}^T$ respectively. The dot product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted either as $\mathbf{x}^T\mathbf{y}$ or $\mathbf{x} \cdot \mathbf{y}$. Functions are indicated with lowercase italic letters, such as f, with $f : A \to B$ mapping between the feature space A and output space B. Pairs like a and b are depicted as tuples (a, b).

1.3.2 Neural Networks and Deep Learning

This section provides a simplified overview of neural networks and deep learning, focusing on applications related to the thesis. For more comprehensive exploration and an overall grasp of deep learning, readers are encouraged to refer to LeCun et al. (2015), and to Goldberg (2016) for insights into deep learning within the context of Natural Language Processing (NLP).

In machine learning, which neural networks are a subset of, there are three main learning paradigms: *supervised*, *unsupervised*, and *reinforcement learning*. Our discussion is framed within the supervised learning paradigm, which underpins most of the neural network-based applications and models used in this work. In supervised learning the model learns from labeled training data, adjusting its parameters to minimize the error in its predictions. The training data consists of input-output pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ represents the input features and y_i represents the corresponding target output; for instance, movie reviews and their corresponding sentiment classes. In a typical supervised learning setting, the dataset is divided into three parts: training, development (or validation), and test datasets. The model is trained on the training dataset, its hyperparameters are adjusted based on the performance on the development dataset, and finally, the model is evaluated on the test dataset.

Mathematically, the goal of supervised learning can be described as learning a function $f : \mathcal{X} \to \mathcal{Y}$ that maps an input $\mathbf{x} \in \mathcal{X}$ to an output $y \in \mathcal{Y}$, by minimizing a loss function $L(y, f(\mathbf{x}))$ over the dataset. This process involves optimizing the parameters θ of the function f to reduce the difference between the predicted output $f(\mathbf{x}; \theta)$ and the actual output y across all input-output pairs.

Neural Networks

At their core, neural networks are computational models inspired by the human brain's architecture, designed to recognize patterns from complex data. A neural network comprises layers of interconnected nodes or neurons, where each connection has an associated weight. Information flows from the input layer to the output layer, possibly through multiple hidden layers, transforming the input through a series of linear and non-linear operations.

Mathematically, neural networks can be described as non-linear mappings $\mathbf{f}_\theta : \mathcal{X} \to \mathcal{Y}$, characterized by a parameter vector $\theta \in \mathbb{R}^k$, and are differentiable with respect to θ. These networks, typically comprising vectors, matrices, or tensors, involve a series of layers sequentially executing operations on the input data, transforming it into a form suitable for the desired output.

Neural Network Layers

While there are many important and diverse types of layers in neural networks, such as convolutional layers (Lecun et al., 1998), recurrent layers (Rumelhart et al., 1986) and LSTM layers (Hochreiter and Schmidhuber, 1997), here we focus on essential types that are relevant to the models used in this work.

Embedding Layers For the applications where the input is not in the form of real-valued vectors, the embedding layer transforms these discrete input tokens from a vocabulary, typically words, characters or subword units in NLP, into continuous vectors. Mathematically, it's represented as a lookup operation in an embedding matrix $\mathbf{E}$, where $\mathbf{E} \in \mathbb{R}^{V \times D}$. Here, V is the size of the vocabulary, and D is the dimensionality of the embeddings. For an input token with index i, the embedding vector is obtained as $\mathbf{e}_i = \mathbf{E}_{i,\cdot}$. This operation projects sparse, high-dimensional categorical input features into a lower-dimensional, continuous space, facilitating subsequent learning tasks in dense format.

Feed-Forward Layers Feed-forward layers, also known as fully connected layers, consist of linear transformations followed by non-linear activations. Given an input vector $\mathbf{x} \in \mathbb{R}^n$, the output $\mathbf{y} \in \mathbb{R}^m$ of a feed-forward layer is calculated as $\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$. Here, $\mathbf{W} \in \mathbb{R}^{m \times n}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^m$ is the bias vector, and $\sigma(\cdot)$ represents the non-linear activation function such as ReLU or Sigmoid. Each node in a layer connects to all nodes in the preceding and subsequent layer, which makes these layers "fully connected".

Attention Layers Originally introduced for neural machine translation (Bahdanau et al., 2014), the attention mechanism allows the model to focus on different parts of the input sequence when predicting an output sequence, essentially weighting the input's significance dynamically. Consider two sequences of k-dimensional vectors, $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_m)$ and $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$, where each $\mathbf{q}_i \in \mathbb{R}^k$ and $\mathbf{v}_j \in \mathbb{R}^k$ for $1 \leq i \leq m$ and $1 \leq j \leq n$. In this context, the elements of $\mathbf{q}$ are referred to as queries and those of $\mathbf{v}$ as values. The attention mechanism maps queries $\mathbf{q}$ to a new sequence of vectors $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_m)$ where each $\mathbf{w}_i$ is derived as a weighted sum of all vectors $\mathbf{v}_j$, specifically:

$$\mathbf{w}_i = \sum_{j=1}^{n} \alpha_{i,j} \mathbf{v}_j$$

where $\alpha_{i,j}$ represents the weight assigned to the j-th value when computing the new representation for the i-th query.

In their ground-breaking “Attention is All You Need" paper, Vaswani et al. (2017) proposed computing the attention weights using *scaled dot-product attention*. Using the matrix notation for the sequences of vectors, queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$, the attention weights are computed using the formula:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

The scaling factor $\sqrt{d_k}$, where d_k is the dimensionality of the keys, helps prevent the dot products from growing too large in magnitude, leading to smoother gradients during training. This scaled attention mechanism ensures that the model can dynamically prioritize which inputs are most relevant for producing each output element.

Self-attention, is a variant of the attention layer that allows the model to relate different positions of a single sequence in order to compute a representation of the sequence itself. In self-attention, the queries, keys, and values all come from the same input sequence, transformed into different representations. To allow the model to jointly attend to information from different representation subspaces at different positions, the concept of *multi-headed attention* is utilized. Instead of performing a single attention operation, the input is linearly transformed multiple times with different learned projections to queries, keys, and values. This results in multiple sets of queries, keys, and values, over which scaled dot-product attention is independently computed. The outputs of these independent attention operations are then concatenated and linearly transformed into the expected dimensionality. The multi-headed attention mechanism can be expressed as:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O$$

where each head i is computed as:

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

and $\mathbf{W}^O$ is the output projection matrix.

Each $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ are parameters to be learned, with d_k, d_v typically being d_{model}/h. These projections into d_k, d_v are meant to decrease the dimensionality for each head, allowing for efficiency and robustness in learning different aspects of the data in different subspaces.

Softmax Layers The softmax layer is typically used as the final layer of a neural network model for classification tasks. It transforms the raw output scores (logits) from the network into probabilities by taking the exponential of each output and then normalizing these values by dividing by the sum of all the exponentials. Mathematically, for a vector of logits $\mathbf{z} \in \mathbb{R}^K$ representing K classes, the softmax function $\sigma(\mathbf{z})_i$ for each class i is given by:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$$

where z_i is the logit corresponding to the i-th class. This transformation ensures that the output values are non-negative and sum up to 1, making them interpretable as probabilities.

Training Neural Networks

Training neural networks involves adjusting the weights of connections between neurons to minimize the loss (or cost) function, which measures the difference between the actual and predicted outputs. A loss function, $L(y, \hat{y})$, measures the discrepancy between the true labels y and the predicted labels $\hat{y}$ by the model. The choice of the loss function depends on the specific task at hand. For example, for regression tasks, where the goal is to predict continuous values, the *Mean Squared Error* (MSE) loss function is commonly used:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

For classification tasks, where the goal is to assign each input into one of several classes, the *Cross-Entropy* loss is often used:

$$\text{Cross-Entropy} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c})$$

where N is the number of samples, C is the number of classes, $y_{i,c}$ is a binary indicator of whether class c is the correct classification for observation i, and $\hat{y}_{i,c}$ is the predicted probability that observation i is of class c.

Gradient Descent is the foundational optimization algorithm used in training neural networks. It adjusts the model's parameters θ iteratively to minimize the loss function. The model parameters are updated by moving in the negative direction of the gradient of the loss function with respect to the parameters. The *learning rate*, α, is a critical hyperparameter that controls the size of the steps taken towards the minimum of the loss function. The update rule can be represented as:

$$\theta_{\text{new}} = \theta_{\text{old}} - \alpha \nabla_{\theta} L(\theta)$$

Choosing the appropriate learning rate is crucial; too small a learning rate leads to slow convergence, while too large a learning rate can cause the optimization to overshoot the minimum or even diverge.

Advanced optimization algorithms like Stochastic Gradient Descent (SGD), Adam, and RMSprop have been developed to address the limitations of the basic gradient descent algorithm, particularly in terms of speed of convergence and the ability to escape local minima. These optimizers introduce concepts such as momentum and adaptive learning rates to improve the training dynamics.

1.3.3 Deep Learning for NLP

When applying deep learning to natural language processing (NLP) tasks, there are many important details to consider. Here we discuss three key areas that are most prominent for this thesis. First, we explore *tokenization*, which involves breaking down text into individual tokens. Then, we discuss the *Transformer* architecture, a type of neural network architecture introduced by Vaswani et al. (2017). This architecture has been empirically demonstrated to scale very well and achieve impressive results across a broad spectrum of NLP tasks, as highlighted by numerous studies (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020). We specifically focus on the transformer architecture as it currently dominates the field of NLP[1] and is the basis for many of the models used in this work.

1.3.3.1 Tokenization

When feeding text data into neural networks, a common first step, even before the embedding layer, is to tokenize the text into smaller units, called tokens. This process, known as *tokenization*, is crucial for transforming raw text into a structured form that can be processed by the model. Tokenization techniques can be broadly classified into three categories: *word*, *character*, and *subword* tokenization. Each approach has its strengths and has been applied successfully in various NLP tasks. Below we discuss the techniques of tokenization in more detail.

[1] Transformers is becoming the dominant architecture in other AI fields as well as of the writing of this thesis

Word-Level Tokenization Word-level tokenization involves segmenting text into words based on spaces and punctuation. This technique is straightforward and intuitive, aligning well with the human understanding of text. However, one important problem with word tokenization is it that it does not allow the model to leverage the apparent similarities in form with other terms. For instance, when functioning at the word level, a neural network cannot deduce the meanings of terms like "bakery" or "advertisement" from the words "bake" and "advertise" since the model processes them completely independently. Consequently, models that employ word-level tokenization fail to provide meaningful interpretations to words not encountered during their training, as they lack a method to assimilate information about these unfamiliar words. To address this, such models typically employ a unique symbol (such as $\langle UNK \rangle$) to denote all unknown terms. Moreover, word tokenization scales poorly with the size of the vocabulary, as the number of unique words in a language can be vast, especially in morphologically rich languages, leading to computational inefficiencies.

Character-Level Tokenization Character-level tokenization decomposes text into individual characters. This approach is language-agnostic and allows for a smaller vocabulary size, as the number of unique characters is significantly smaller than the number of unique words. Character-level tokenization also allows models to handle rare or unseen words by breaking them down into familiar characters, thus mitigating the risk of encountering out-of-vocabulary (OOV) words. However, character-level tokenization can lead to models that require more computational resources, as the sequences they process are significantly longer than word-level sequences.

Subword-Level Tokenization To address the limitations of the word and character-level tokenization, various subword tokenization techniques, such as *Byte Pair Encoding* (BPE) (Sennrich et al., 2016), *WordPiece* (Wu et al., 2016), and *SentencePiece* (Kudo and Richardson, 2018), have been developed. The key idea behind subword tokenization is to segment words into smaller units using the frequency of character sequences such that the frequent words are kept intact while the rare words are broken down into subword units (or into individual characters in the worst case). This technique is shown to be effective in handling rare and unseen words, while also reducing the size of the vocabulary, thus improving the computational efficiency of the model.

1.3.3.2 Transformers Architecture

Before the advent of transformers, the dominant architectures in NLP were recurrent neural networks (RNNs) (Rumelhart et al., 1986), long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997), and gated recurrent units (GRUs) (Cho et al., 2014). These models processed data sequentially, which inherently limited parallelization, making training on large datasets computationally expensive. Moreover, RNNs and their variants struggled with long-range dependencies due to vanishing or exploding gradient problems.

To address these limitations, researchers experimented with various attention mechanisms, initially as a supplementary component to enhance RNNs (Bahdanau et al., 2014). The transformer model (Vaswani et al., 2017) was the first to successfully use a self-attention mechanism as the central architecture, entirely replacing recurrent layers with a fully attention-based approach. Although it has been initially proposed for machine translation, the transformer architecture has since been adapted and extended to a wide range of NLP tasks, achieving state-of-the-art performance on many benchmarks (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020) This novel approach has not only enabled models to capture long-range dependencies more effectively but also significantly improved training time by allowing for greater parallelization and scalability.

The transformer model is based on a simple yet powerful architecture consisting of an *encoder* that transforms the input tokens to contextualized vector representations using self-attention and a *decoder* that generates the output sequence token-by-token by attending to the encoder's output as well as its own previous outputs. Figure 1.1 illustrates the transformer architecture.

Transformer Encoder The encoder consists of a stack of L blocks (also referred as layers). These blocks have an identical structure, each containing two sublayers: a multi-headed self-attention mechanism and a position-wise feed-forward neural network. However, the parameters in each block are unique and learned independently. Let's denote the input to the l-th encoder block as $\mathbf{H}^l \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the dimensionality of each vector in the sequence. The output of the l-th encoder block is another matrix $\mathbf{H}^{l+1} \in \mathbb{R}^{n \times d}$, which becomes the input to the $l + 1$-th encoder block.

Inside an encoder block, the input $\mathbf{H}$ is first passed through a multi-headed self-attention mechanism, which allows the model to weigh the importance of each token in the sequence when computing the representation of each token. Then, a residual connection (He et al., 2015) combines the output of the self-attention mechanism with the input to the block, followed by layer normalization (Ba et al., 2016). Afterward, the output is passed through a feed-forward neural network, which consists of two linear transformations with a non-linear activation function in between. Again, the output of the feed-forward network is combined with the input to the block using a residual connection and normalized using layer normalization, which creates the final output of the encoder block.

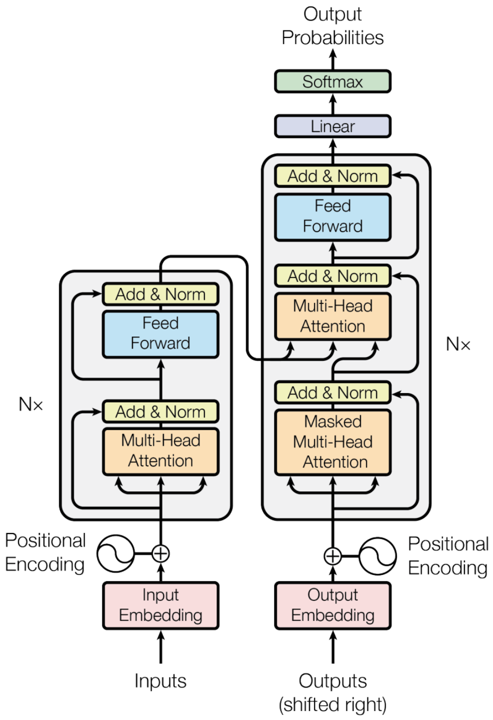


Figure 1.1 – *The Transformer architecture. Figure taken from Vaswani et al. (2017).*

Mathematically, if we denote the function representing the entire encoder block as f_l, we can represent the transformation from input to output for the l-th encoder as:

$$\mathbf{H}^{l+1} = f_l(\mathbf{H}^l)$$

For the first encoder block, the input $\mathbf{H}^0$ would typically be the embeddings of the input tokens (added with *positional encodings*), and the output $\mathbf{H}^1$ would feed into the next block. Hence, we can describe the chaining of the encoder blocks in a transformer as:

$$\begin{aligned}
\mathbf{H}^1 &= f_1(\mathbf{H}^0) \\
\mathbf{H}^2 &= f_2(\mathbf{H}^1) \\
&\vdots \\
\mathbf{H}^{(L)} &= f_L(\mathbf{H}^{(L-1)})
\end{aligned}$$

where L is the number of encoder blocks in the transformer. $\mathbf{H}^{(L)}$ is the final output of the encoder, which contains the contextualized representations of the input tokens.

Transformer Decoder The transformer decoder is similar to the transformer encoder but has two key differences. First, the decoder uses a *masked self-attention* mechanism in its self-attention layer to prevent tokens from attending to future tokens during training by setting the attention weights to zero for future tokens. Second, the decoder has an additional multi-headed attention mechanism that attends to the encoder's output, referred as *cross-attention*, allowing the decoder to focus on different parts of the input sequence when generating the output sequence. Note that the cross-attention is not used in *decoder only* models since they do not have access to the encoder's output. Transformer decoder contains m decoder blocks followed by a linear transformation and a softmax layer to obtain a probability distribution over the target vocabulary in order to predict the next token in the output sequence.

Building on the original transformers architecture, many works have proposed variations and extensions to the different parts of the architecture to improve their efficiency and address some of their limitations. Parameter sharing (Takase and Kiyono, 2023), sparse and efficient attention mechanisms (Dao et al., 2022; Liu et al., 2023, 2022), efficient training and inference strategies (Lepikhin et al., 2021; Yuan et al., 2024) are some of the main modifications to improve the transformer architecture.

Positional Encoding Since, unlike recurrent neural networks (RNNs) or long short-term memory networks (LSTMs), transformers do not inherently process data in sequence, they require a mechanism to incorporate the order of the input data. To address this, transformers use positional encodings (Gehring et al., 2017), which are vectors added to the input embeddings to provide information about the position of tokens in the input sequence. These vectors can be computed in various ways. In the original transformers paper, Vaswani et al. (2017) proposed using a sinusoidal function for generating these vectors, with different frequencies for each dimension. The mathematical formulation is as follows:

For position pos and dimension $2i$ within the vector (where i is an integer), the positional encoding is:

$$\mathbf{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

And for dimension $2i + 1$:

$$\mathbf{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

where:

- $\mathbf{PE}_{(pos,2i)}$ and $\mathbf{PE}_{(pos,2i+1)}$ are the positional encoding values for even and odd dimensions, respectively.
- pos is the position of the token in the sequence.
- d_{model} is the dimension of the token embeddings and positional encoding vectors.
- i represents the current dimension.

This formulation ensures that each position generates a unique encoding. Moreover, because the function is periodic, it allows the model to generalize to sequence lengths unseen during training. However, many of the transformer-based models used in this work, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and GPT-2 (Raffel et al., 2020), employ positional encodings that are randomly initialized and learned during training, rather than using the sinusoidal function used in the original transformer model.

1.3.4 Representation Learning for NLP

Natural Language Processing aims to develop programs that allow machines to understand human languages, which is challenging due to the unstructured nature of language with its multiple granularities (e.g., words, phrases, sentences, documents), tasks (e.g., sentiment analysis, translation), and domains (e.g., news, literary works). Representation learning offers a solution, allowing for the integration of various linguistic elements, tasks, and domains into a unified semantic vector space. This approach not only supports more efficient and robust NLP systems by facilitating knowledge transfer across different levels, tasks, and domains, but also enhances the overall effectiveness of NLP performance.

In this work we focus on methods and models that learn representations for words or subword units, using a *self-supervised* learning approach where the model learns from the data itself without requiring labeled data.

1.3.4.1 Static Representations

Static representations, also known as *static embeddings*, are fixed-size vectors that represent words or subword units in a continuous vector space. These representations are pretrained on large corpora in a self-supervised manner by utilizing the word co-occurrences based on the distributional hypothesis (Harris, 1954; Firth, 1957), which posits that words that appear in similar contexts have similar meanings. Although many different static embedding methods have been proposed, here we limit our discussion to *Word2Vec* (Mikolov et al., 2013b), *GloVe* (Pennington et al., 2014), and FastText (Bojanowski et al., 2017), the three methods that are also most relevant to our work.

Word2Vec Word2Vec (Mikolov et al., 2013b) is a popular method for learning vector representations of words in a continuous vector space. The two main architectures used in Word2Vec are the *Continuous Bag of Words* (CBOW) and *Skip-Gram* models. Here, we will focus on the mathematical formulation of the Skip-Gram model, which predicts context words given a target word and is more widely used due to its better performance on many tasks.

Given a sequence of training words $w_1, w_2, \ldots, w_T$, the objective of the Skip-Gram model is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

where c is the size of the training context (which can be asymmetric). The probability $p(w_{t+j}|w_t)$ of observing a context word w_{t+j} given the current word w_t is defined using the softmax function:

$$p(w_O|w_I) = \frac{\exp(\mathbf{v}_{w_O}^T \mathbf{u}_{w_I})}{\sum_{w=1}^{W}\exp(\mathbf{v}_w^T \mathbf{u}_{w_I})}$$

Here, $\mathbf{u}_{w_I}$ and $\mathbf{v}_{w_O}$ are the "input" and "output" vector representations of the words w_I and w_O, respectively. W is the vocabulary size, and $\mathbf{v}_w$ is the output vector for any word w.

The objective function involves a computationally expensive calculation of the denominator in the softmax function over the entire vocabulary for each training instance. To address this, techniques such as *negative sampling* or *hierarchical softmax* are used.

Negative sampling modifies the objective function by sampling negative words (words not in the context). For each pair of words (w_I, w_O), where w_O is a context word of w_I, we sample k negative words not in the context. The modified objective function is:

$$\log \sigma(\mathbf{v}_{w_O}^T \mathbf{u}_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-\mathbf{v}_{w_i}^T \mathbf{u}_{w_I})\right]$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function, and $P_n(w)$ is a noise distribution, typically chosen as the unigram distribution raised to the 3/4 power.

During training, for each word w_t in the corpus, the model updates the vectors $\mathbf{u}_{w_t}$ and $\mathbf{v}_{w_{t+j}}$ for each context word w_{t+j} and each negative sample by ascending on the gradient of the log probability of the correct classification of w_{t+j} as a context word of w_t and the negative samples as non-context words.

GloVe GloVe (Pennington et al., 2014) is another widely used static embedding method that learns word representations by factorizing the word co-occurrence matrix. One key difference of the GloVe method from word2vec is that it uses global word-word co-occurrence statistics from a corpus, contrary to the Word2Vec which uses local context information within a sliding window. GloVe model is trained to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Given a corpus, we first construct a co-occurrence matrix $\mathbf{X}$ where $\mathbf{X}_{i,j}$ is the number of times word j occurs in the context of word i. Let $\mathbf{w}_i$ and $\mathbf{w}_j$ be the word vectors for words i and j, respectively. Additionally, each word i has a bias b_i and each word j has a bias b_j.

The GloVe model minimizes the following objective function:

$$J = \sum_{i,j=1}^{V} f(\mathbf{X}_{i,j}) \left(\mathbf{w}_i^T \mathbf{w}_j + b_i + b_j - \log \mathbf{X}_{i,j}\right)^2$$

where V is the vocabulary size, and f is a weighting function that assigns lower weight to rare and frequent co-occurrences. A common choice for f is:

$$f(x) = \begin{cases} (x/x_{\text{max}})^\alpha & \text{if } x < x_{\text{max}} \\ 1 & \text{otherwise} \end{cases}$$

with $\alpha \approx 0.75$ and x_{max} typically set to 100.

FastText FastText (Bojanowski et al., 2017) is an extension of the Word2Vec model which incorporates subword information by considering character n-grams in addition to whole words. This approach was proposed to capture the morphological structure of words, especially useful for languages with rich morphology and for handling out-of-vocabulary (OOV) words.

For a given word w, FastText considers not only the word itself but also its constituent character n-grams. Let G_w be the set of n-grams for word w. For example, for the word "cat" and n-gram sizes of 3, G_{cat} might include the n-grams: "<ca", "cat", "at>" (where "<" and ">" are boundary symbols indicating the start and end of the word).

Each n-gram g has an associated vector $\mathbf{z}_g \in \mathbb{R}^d$. The word vector $\mathbf{v}_w$ for the word w is computed as the sum of the vectors of its n-grams:

$$\mathbf{v}_w = \sum_{g \in G_w} \mathbf{z}_g$$

FastText uses a variant of the skip-gram model, where it predicts context words based on the current word.

1.3.4.2 Contextualized Representations

One of the main shortcomings of static embeddings is that they do not capture the context in which a word appears, leading to the same representation for a word regardless of its meaning in different contexts. For instance, the word “bank" has different meanings in the context of “river bank" and “financial bank", but static embeddings would assign the same vector to both instances, failing to capture the word’s contextual meaning. To address this limitation, contextualized embeddings has been proposed by the *ELMo (Embeddings from Language Models)* model (Peters et al., 2018) where *language model pretraining* is used to learn contextualized representations of words. The main idea behind the language model pretraining is to train a model to predict the next word in a sequence given the previous words, which forces the model to obtain a deep knowledge and understanding about the syntax and semantics of the language. This knowledge is stored in the model’s parameters, which can generate context dependent representations for words that can be transferred to downstream tasks. Although the learned contextual representations were kept fixed after the pretraining phase in ELMo, the models that followed, such as *ULMFiT* (Howard and Ruder, 2018), *GPT* (Radford and Narasimhan, 2018), and *BERT* (Devlin et al., 2019) popularized the approach of *fine-tuning* these representations on the downstream tasks using labelled task specific data, which further improved the performance of the models. Since the self-supervised language model pretraining only requires unlabelled textual data, which is abundant and easy to collect for most languages, the approach allows for the transfer of knowledge from large-scale pretraining to a wide range of downstream tasks, making it a powerful tool for NLP.

Below we discuss the main language model pretraining methods that have been proposed in the literature and provide specific examples of models that have been developed based on these methods.

Masked Language Modeling

In order to learn contextualized representations of words by using their contexts on both sides of the word, Devlin et al. (2019) introduced the *masked language modeling* (MLM) objective. The key idea behind this objective is to mask a subset of the input tokens and train the model to predict the masked tokens based on the surrounding tokens. The masking is done by replacing some tokens in a sequence with a special token (i.e., `[MASK]`) and training the model to predict the original tokens using the contextualized representations of the special masked tokens. Although different masking ratios have been suggested, the most common choice is to mask 15% of the tokens in the input sequence as proposed by Devlin et al. (2019).

Mathematically, given a sequence of tokens $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ where each token s_i is an element from a vocabulary $\mathcal{V}$, a subset of tokens in s is randomly selected and replaced with a special mask token. Let $\mathbf{m}$ denote the masked version of $\mathbf{s}$, where any token s_i that is masked is replaced by the mask token.

Each token m_i in $\mathbf{m}$ is mapped to a vector $\mathbf{e}_i \in \mathbb{R}^d$ using an embedding function $\text{emb} : \mathcal{V} \to \mathbb{R}^d$. Thus, $\mathbf{e}_i = \text{emb}(m_i)$. A neural network model (e.g., transformer encoder) processes the sequence of embeddings $(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)$ to produce contextualized representations $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$, where $\mathbf{h}_i \in \mathbb{R}^d$. For each masked token m_i, the model predicts the original token s_i using a prediction function f (e.g., a linear layer followed by a softmax function). The function f maps the contextualized representation $\mathbf{h}_i$ to a probability distribution over the vocabulary $\mathcal{V}$, i.e., $f(\mathbf{h}_i) \in \mathbb{R}^{|\mathcal{V}|}$, where $|\mathcal{V}|$ is the size of the vocabulary.

The model is trained to minimize the loss between the predicted probability distribution and the actual token. Typically, this is done using cross-entropy loss (see Section 1.3.2). The parameters of the embedding function emb, the neural network, and the prediction function f are optimized to minimize the total loss over all masked tokens in a training dataset.

Masked language models in general use the encoder part of the transformer architecture to generate contextualized representations of the tokens in the input sequence which enables them to utilize both left and right context of the tokens to generate the representations. The obtained contextual representations are shown to perform well for many classification tasks, especially after task specific finetuning of the models. However, models trained with masked language modeling are not well suited for text generation tasks as they do not have the autoregressive property to generate text token by token. Below we discuss some of the popular models that have been developed based on the masked language modeling objective.

BERT BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is the first model to introduce the masked language modeling objective for pretraining transformer models. However, it contains some differences from the pure masked language modeling objective described above. First, among the input tokens that are selected for masking, BERT only replaces 80% of them with the mask token while 10% of them are replaced with a random token from the vocabulary, and remaining 10% are left unchanged. Moreover, in addition to the masked language modeling objective, BERT also uses a *next sentence prediction* (NSP) objective where the model is trained to predict whether two sentences are consecutive in the original text. This is done by adding a special classification token (i.e., `[CLS]`) at the beginning of each input sequence and a binary classification head on top of the contextualized representation of this token to predict whether the second sentence follows the first sentence in the original text. A special token `[SEP]` is used to separate the two sentences in the input sequence. During finetuning on a classification task, the output of the `[CLS]` token is used as the representation of the entire input sequence and fed into a task-specific classifier. Bert is pretrained on combination of a Wikipedia dump and the BooksCorpus (Zhu et al., 2015) as two variants: BERT-base and BERT-large, with 110M and 336M parameters, respectively.

RoBERTa RoBERTa (Robustly Optimized BERT Pretraining Approach) (Liu et al., 2019) is an extension of BERT that modifies the training procedure and hyperparameters to improve performance. Key differences of RoBERTa from BERT include: (i) removing the NSP objective which is found to be not helpful, (ii) using dynamic masking during training instead of static masking used by BERT, and (iii) training with larger batches for longer on more data. It is shown that RoBERTa significantly outperforms BERT on many benchmarks thanks to these modifications.

XLM-R XLM-R (Cross-lingual Language Model - RoBERTa) (Conneau et al., 2020) is a multilingual variant of RoBERTa that is pretrained on a large corpus of 100 languages without using any parallel data.

ALBERT ALBERT (Lan et al., 2020) is a version of BERT that aims to reduce model size and increase training speed without significantly affecting performance. The key differences of ALBERT from BERT include: (i) using a factorized embedding parameterization that reduces the size of the vocabulary, (ii) sharing parameters across layers which reduces the memory footprint of the model, and (iii) using a sentence order prediction (SOP) objective instead of the NSP objective used by BERT.

DistilBERT DistilBERT (Sanh et al., 2019) is a smaller, faster, and lighter version of BERT. It has about 40% fewer parameters than BERT, runs 60% faster, and retains over 95% of BERT's performance in most tasks. In order to achieve this, DistilBERT is trained using knowledge distillation where it learned to mimic the behavior of a larger teacher model (BERT) during training.

Sequence-to-Sequence Language Modeling

The goal of *Sequence to sequence* (seq2seq) language modeling is to transform a given sequence of elements (words, characters, etc.) in one language into another sequence in the same or a different language. In general, seq2seq models include both the encoder that processes the input sequence and the decoder that generates the output sequence. Due to their architecture, seq2seq models are well suited for both classification and generation tasks, making them versatile for a wide range of NLP tasks. Before the introduction of pretraining by Devlin et al. (2019), seq2seq models, including the original transformers model (Vaswani et al., 2017), were commonly trained directly on supervised datasets for specific tasks like machine translation, summarization, and question answering.

BART After the popularization of the self-supervised pretraining approach with BERT, Lewis et al. (2020) introduced BART, a denoising autoencoder for pretraining sequence-to-sequence models. Bart is trained by corrupting the input sequence with an arbitrary noising function and then training the model to reconstruct the original sequence. The key difference from masked language modeling is that the noising function can be more complex and usually involves sequences of tokens rather than individual tokens. In their work, Lewis et al. (2020) experimented with various denoising strategies and found the best performing strategies to be: (i) randomly shuffling the order of the input sentences, and (ii) replacing spans of text with a single mask token.

T5 The T5 (Text-to-Text Transfer Transformer) model (Raffel et al., 2020) is a popular seq2seq model that is pretrained by replacing spans of text with a single mask token and training the model to generate the original text. In addition to the unsupervised pretraining on large unlabeled corpora, T5 is also trained on multiple supervised datasets where the tasks are converted to a text-to-text format to enable the model to perform a wide range of tasks using the same architecture.

Autoregressive Language Modeling

Consider a sequence of words $\mathbf{w} = (w_1, w_2, \ldots, w_T)$, where each w_t is a word from a vocabulary $\mathcal{V}$. The goal of autoregressive language modeling is to estimate the probability of the sequence $\mathbf{w}$, which can be decomposed using the chain rule of probability as follows:

$$P(\mathbf{w}) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\cdots P(w_T|w_1, w_2, \ldots, w_{T-1})$$

Each term $P(w_t|w_1, w_2, \ldots, w_{t-1})$ represents the conditional probability of the word w_t given all the previous words $w_1, w_2, \ldots, w_{t-1}$. An autoregressive language model, such as LSTMs (Peters et al., 2018) or a transformer decoder (Radford and Narasimhan, 2018; Radford et al., 2019), is trained to maximize the likelihood of the next word in the sequence given the previous words. The task of predicting the next word in the sequence is typically done by using a softmax function over the vocabulary to predict the probability distribution of the next word, where the cross-entropy loss is used to train the model. The next word prediction task makes the autoregressive language models inherently suitable for text generation tasks since they match the pretraining objective of the models. One disadvantage of autoregressive language modeling is that the contextual representations of a word can only incorporate the left context of the word, which may limit the model's ability to capture the full context of the word. In order to address this, Peters et al. (2018) proposed training one left-to-right and one right-to-left model and combining their output representations to obtain a better contextual representation of the word.

Most of the current state-of-the-art large language models are based on autoregressive language modeling. Below we discuss some of the popular autoregressive language models that have been developed before the rise of large language models and we investigate the recent large language models in the next section.

XL-Net XL-Net (Yang et al., 2019) is a language model that is based on Transformer decoder and trained using an autoregressive language modeling objective. However, in order to make use of the contexts bidirectionally, XLNet uses a *permutation language modeling* objective where the model is trained to predict the next word in a sequence given the previous words, but the previous words are not necessarily the words that appear before the current word in the sequence. Instead, the model is trained to predict the next word given a permutation of the input sequence, which allows the model to capture bidirectional context information. Yang et al. (2019) showed that XLNet outperforms BERT on many benchmarks by using this permutation language modeling objective.

GPT and GPT-2 Radford and Narasimhan (2018) first introduced the *generative pretraining* in their work where they pretrained the first GPT model on BooksCorpus (Zhu et al., 2015) and then finetuned on several downstream tasks. Later, Radford et al. (2019) extended and popularized the generative pretraining approach with the GPT-2 model, which is pretrained on a larger dataset and showed state-of-the-art performance on many benchmarks.

1.3.4.3 Large Language Models

The current landscape of NLP is dominated by large language models (LLMs), which are fundamentally based on the transformer architecture and pretrained using self-supervised learning on massive corpora. While there is no clear boundary defining how many parameters constitute a large language model, models with at least several billion parameters are generally considered large. LLMs are created by scaling existing transformer models and their pretraining corpora to larger sizes. GPT-3 (Brown et al., 2020), developed by OpenAI, is often considered the first breakthrough in the realm of large language models, featuring an unprecedented 175 billion parameters. Below we discuss important topics related to the training and application of large language models. Although these topics are not closely related the most of the work done in this thesis, they are important to understand the current landscape of NLP and the challenges and opportunities that large language models, such as chatGPT which is utilized in this thesis, bring to the field.

Scaling Laws One of the main motivations for scaling the size of language models is the observation of *scaling laws* in machine learning, where increasing the model size and the amount of training data leads to improved performance on various tasks. This phenomenon is often referred to as the "bitter lesson" in AI research (Sutton, 2019), where it is observed that larger models trained on more data tend to perform better, regardless of the specific architectural choices. In their work Kaplan et al. (2020) showed that as the language models get larger, in addition to becoming more performant, they also become more data-efficient, requiring fewer examples to reach a certain level of performance.

In-Context/Few-Shot Learning One of the abilities that large language models have demonstrated is *in-context* or *few-shot learning*. In-context or few-shot learning is a paradigm where the model leverages a few examples to understand and execute new tasks, effectively learning from the context provided in the input prompt. This capability is revolutionary because it bypasses the traditional machine learning requirement of training on large, labeled datasets for each new task. For example, GPT-3 can generate summaries, answer questions, or translate languages with minimal to no specific training on these tasks, relying instead on its general pretraining. The effectiveness of in-context learning is primarily due to the distributed representation of knowledge across the model's many parameters, which allows it to flexibly apply this knowledge to a wide range of tasks. However, this method is not without limitations, as the quality of the outcomes can vary significantly based on how the prompts are structured and the specific examples provided (Köksal et al., 2023).

Prompting Since the language models are mainly interacted with using text prompts, the quality of the prompts is crucial for obtaining the desired outputs from the models. Many studies showed that the output of the language models can be significantly influenced by the choice of the prompts (Chen et al., 2023; Gonen et al., 2023; Sclar et al., 2023), and particular prompting strategies such as *chain-of-thought prompting* (Wei et al., 2023; Wu et al., 2023) can significantly improve the performance of the models on various tasks.

Instruction Tuning Language models that are trained only on large corpora often struggle with following specific instructions or generating outputs that align with intentions of the humans who interact with them. This is because a model that is trained with next word prediction objective on generic text data learns the most likely continuation for its prompt based on the statistical patterns in the data, and web text is not structured in a way that is conducive to following specific instructions. To address this issue, *instruction tuning* is proposed which is a method where the model is fine-tuned on a dataset structured around following specific instructions. Several studies focused on creating instructions to solve common NLP tasks (Wei et al., 2022; Sanh et al., 2022; Wang et al., 2022; Honovich et al., 2023) and these instructions are usually used to finetune large sequence-to-sequence language models such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020). Several other studies explored instruction-based approaches in more general settings (Ouyang et al., 2022; Wang et al., 2023; Köksal et al., 2024); others explored instruction-based approaches in more general settings (Li et al., 2023a; Sclar et al., 2023; Ouyang et al., 2022). Although instruction tuning significantly enhanced the practical usability of LLMs in real-world applications, where users can interact with the model using natural language instructions without needing to understand the underlying model complexities or having to provide numerous examples, the models are still shown to be sensitive to how the users phrase their instructions (Li et al., 2023a; Sclar et al., 2023).

Reinforcement Learning from Human Feedback As the large language models become more capable, it became increasingly important to ensure that the outputs generated by these models are aligned with human values and preferences, in order to prevent harmful or inappropriate content from being generated. *Reinforcement Learning from Human Feedback* (RLHF) is a technique proposed by Christiano et al. (2017) in order to solve complex RL tasks without access to the reward function by providing feedback in the form of human preferences between pairs of actions or outputs for only a small fraction of the agent's interactions. This technique is applied to LLMs in (Ouyang et al., 2022) by first training a reward model to predict human preferences between pairs of model outputs and then utilizing *Proximal Policy Optimization* (PPO) Algorithms (Schulman et al., 2017) to update the model's parameters based on the reward model's predictions. RLHF became a popular method to improve the safety and appropriateness of LLM outputs. Later techniques such as *Direct Preference Optimization* (DPO) (Rafailov et al., 2023) are proposed to further improve the efficiency and stability of RLHF by directly optimizing the model's outputs to match human preferences.

LLM Landscape The landscape of large language models is rapidly evolving, with new models being introduced frequently, each pushing the boundaries of model size, performance, and capabilities. In recent years, we witnessed a significant shift from open-source models to proprietary, closed-source systems where the access to the models is only available through APIs. This transition is largely driven by the financial incentives because of the escalating costs and complexities associated with training state-of-the-art models, as well as the competitive advantage that these models provide to the organizations that develop them.

Another important trend in the LLM landscape is the increasing focus on the high quality data, both for the pretraining and supervised fine-tuning of the models as these models rely heavily on the diversity, size, and accuracy of their training datasets to generate reliable and high quality outputs. While some companies hire human annotators to create high-quality datasets as well as collecting human feedback (Stiennon et al., 2020), others rely on semi-supervised approaches to create useful finetuning datasets from publicly available internet data (Köksal et al., 2024). There is also an increasing focus on investigating to what extent specific synthetic data that is generated by the large language models can be used to train newer models more efficiently (Mitra et al., 2023; Li et al., 2023b).

Looking towards the future, the field of LLMs is poised to evolve into increasingly multimodal domains, integrating text with other data types such as images, videos, and audio. This evolution will enable more comprehensive and interactive AI systems, capable of understanding and generating information across various forms of media, thereby expanding their applicability in real-world scenarios and enhancing user interactions. As these multimodal capabilities advance, they promise to redefine the boundaries of what AI systems can achieve, making them more versatile and integral to digital communication and information processing.

Popular LLM Families

As the LLM landscape is rapidly evolving, new models are introduced frequently, each pushing the boundaries of model size, performance, and capabilities. Below we discuss some of the popular LLM families that have been developed in recent years.

GPT OpenAI continued their scaling of generative transformer models GPT and GPT-2 with their 175 billion parameter GPT-3 model (Brown et al., 2020), which demonstrated and popularized the in-context learning capabilities of large language models. However, the interest in LLMs within NLP community, as well as general public, increased drastically after the release of GPT-3.5-Turbo (also referred as *ChatGPT*), that is followed by the release of GPT-4 (OpenAI et al., 2024). Not much is known (beyond rumors and leaks) about the training data, training strategy and model size of these commercial models and they can only be accessed through APIs or OpenAI's website[2]. As of today, GPT-4 versions remain one of the best performing LLMs in the field.

Gemini One of the leading models in the LLM landscape is the Gemini series of model developed by Google Research (Team et al., 2024) that succeeded their PALM series of large language models (Chowdhery et al., 2024). These models are trained to be multimodal from the start, and their largest versions are deemed to be compatible with the GPT-4 models. Similar to the GPT models, the Gemini models are also not open source and not much is known about their training data and strategies.

Claude Another group of popular large language models that are considered to be on par with GPT-4 and Gemini models are the Claude series models developed by Anthrophic. The largest version of their new 3 series models, Claude-3-Opus, is arguably the strongest model in the field as of the writing of this dissertation as shown by its superior performance on various benchmarks (Anthropic, 2024).

Llama Contrary to the models mentioned above, Llama series models from Meta (Touvron et al., 2023) are open-source and can be accessed through Hugging Face's model hub[3]. The most current version of the Llama models, Llama-3, is available in various sizes (8 and 70 billion parameters) that are trained on 10 Trillion tokens using publicly available data. The Llama models are often fine-tuned by other researchers on various tasks and datasets to improve their performance and

[2] https://chat.openai.com/

[3] https://huggingface.co/models

instruction following capabilities.

Mistral and Mixtral Mistra-7b (Jiang et al., 2023) is a 7 billion parameter open source model developed by MistralAI that is shown to perform better than the comparable sized Llama models. MistralAI also released Sparse Mixture of Experts (SMoE) language model called Mixtral (Jiang et al., 2024) that allows for more efficient inference thanks to Mixture of Experts architecture.

1.3.4.4 Graph Neural Networks

Graph Neural Networks (GNNs) (Scarselli et al., 2009) are a class of deep learning models designed to perform inference on data represented as graphs. They are particularly effective in capturing the dependencies of graph-structured data through the use of node features and edge relationships. At a high level, GNNs are useful for: (i) Node classification, where they determine the category of a node based on its features and the graph structure; (ii) Link prediction, where they predict the existence/likelihood of an edge between two nodes; (iii) Graph classification, where they classify entire graphs based on their structure and node features; and (iv) Node regression, where they predict a continuous value for a node based on its features and the graph structure. In NLP, for a large variety of problems, such as semantic parsing (Sorokin, 2021), text classification (Zhang et al., 2022; Gu et al., 2023) and information extraction (Yu et al., 2021), data can be naturally represented as graphs, making GNNs a powerful tool for these tasks.

GNNs update the representation of a node by aggregating features from its neighbors, iteratively refining node features to capture both local and global graph structures. Although there are many different types of GNNs, such as Graph Attention Networks (GATs) (Veličković et al., 2017), GraphSAGE (Hamilton et al., 2017), and Dynamic Graph Networks (DyGNN) (Trivedi et al., 2019), here we will focus on the most popular type of GNNs, *Graph Convolutional Networks* (GCNs) (Kipf and Welling, 2017), which is also most relevant to this dissertation.

Graph Convolutional Networks GCNs generalize the convolution operation from traditional data (like images) to graph data. They aggregate information from a node's neighbors using a convolution-like operation. This helps in learning a node representation that captures both its features and the topology of the graph. The basic operation in a GCN involves updating the node features by aggregating features from their neighboring nodes, typically using the following formula:

$$\mathbf{H}^{(l+1)} = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$$

where $\mathbf{H}^{(l)}$ is the matrix of node features at layer l, $\mathbf{W}^{(l)}$ is the weight matrix for layer l, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix of the graph $\mathbf{A}$ with added selfconnections $\mathbf{I}$, $\hat{\mathbf{D}}$ is the degree matrix of $\hat{\mathbf{A}}$, σ is a non-linear activation function, such as ReLU.

One specific type of graph convolutional network that is used in this work is *Variational Graph Auto-Encoders* (VGAEs) (Kipf and Welling, 2016) which uses variational auto-encoders to generate latent representations of nodes in a graph. VGAEs are particularly useful for tasks like link prediction and node clustering. The VGAE consists of two parts: an encoder and a decoder. The encoder maps nodes to a latent space:

$$\mathbf{Z} = \mu(\mathbf{X}, \mathbf{A}) + \sigma(\mathbf{X}, \mathbf{A}) \odot \epsilon$$

where:

- $\mu(\mathbf{X}, \mathbf{A})$ and $\sigma(\mathbf{X}, \mathbf{A})$ are the mean and standard deviation of the latent variables, computed by GCNs,

- ϵ is a random noise vector, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$,

- $\odot$ denotes element-wise multiplication.

The decoder reconstructs the graph's adjacency matrix from the latent representations:

$$\hat{\mathbf{A}} = \sigma(\mathbf{Z}\mathbf{Z}^T)$$

where σ is typically the sigmoid function, ensuring the output values are between 0 and 1, suitable for reconstructing a binary adjacency matrix.

VGAEs are trained by maximizing the variational lower bound, which involves a reconstruction loss (e.g., cross-entropy between $\hat{\mathbf{A}}$ and $\mathbf{A}$) and a regularization term derived from the Kullback-Leibler divergence between the approximate posterior of the latent variables and their prior distribution.

1.3.5 Evaluation of NLP Models

Evaluation of NLP models is a crucial aspect of the research process, as it provides insights into the model's performance, generalization capabilities, and suitability for specific tasks. Availability of challenging benchmarks is essential for the progress of the field, as it allows for fair comparison of different models and methods. Here, we distinguish a *task*, a *dataset* and a *benchmark* as follows:

- A *task* is a specific problem that the model is required to solve.

- A *dataset* is a collection of examples that are used to train, validate, and test the model on a specific task.

- A *benchmark* is a standardized evaluation protocol that defines the task, the dataset, the evaluation metric, and the baseline models that are used to compare the performance of the models. A benchmark can often include multiple datasets and tasks. Several benchmarks can also be combined to form a larger benchmark that cover a wide range of tasks and datasets.

Two main evaluation approaches for NLP models (or machine learning models in general) are *intrinsic* and *extrinsic* evaluation. Intrinsic evaluation assesses the performance of a model based on the model's accuracy in executing specific tasks that are fundamental to the model itself, without considering the final application in which the model will be used. Examples of intrinsic evaluation tasks for NLP include measuring the quality of word embeddings, evaluating the performance of a language model on a language modeling task, or probing of language model representations for linguistic properties. The main advantage of intrinsic evaluation is that it directly measures the capabilities and limitations of specific components of an NLP system, providing detailed insights into the model's strengths and weaknesses in controlled settings. However, intrinsic evaluation may not always correlate well with the model's performance on real-world tasks, as it does not consider the complexity and variability of the tasks that the model will encounter in practice (Chiu et al., 2016). Extrinsic evaluation, on the other hand, assesses the model's performance on downstream tasks that are relevant to real-world applications. It evaluates how well the model contributes to the overall goal of a larger system. Examples include tasks like machine translation, document classification, sentiment analysis, or information retrieval. The advantage of extrinsic evaluation is that it provides a clear indication of how useful a model is in real-world applications, reflecting its practical utility and impact.

As the field of NLP has advanced the models have become more complex and capable, the evaluation tasks and benchmarks have also evolved to reflect the increasing complexity and diversity of the tasks that the models are expected to perform. Below we discuss some of the tasks and benchmarks that are relevant to this dissertation as well as some other popular benchmarks and trends in the evaluation of NLP models.

Word Similarity *Word semantic similarity* is a popular intrinsic evaluation task for word embeddings that measures the similarity between pairs of words based on their embeddings and compares it to human judgments of word similarity. The performance of models is evaluated using correlation metrics like Pearson correlation or Spearman's rank correlation. Some of the popular datasets for word similarity evaluation include WordSim-353 (Finkelstein et al., 2001), SimLex-999 (Hill et al., 2015), and SimVerb-3500 (Gerz et al., 2016).

Word Analogy Another popular intrinsic evaluation task for word embeddings is *word analogy*. In the word analogy task, the goal is to complete an analogy of the form: *A is to B as C is to ?*. For example, for the analogy *man is to king as woman is to ?*, the expected answer is “queen". In the word embedding space, this task is evaluated using arithmetic operations on word embeddings, such as checking if the closest word to the vector $e_{\text{king}} - e_{\text{man}} + e_{\text{woman}}$ is “queen". Popular datasets for word analogy evaluation include the Google Analogy Test Set (Mikolov et al., 2013b) and BATS (Bigger Analogy Test Set) (Gladkova et al., 2016).

GLUE and SuperGLUE *The General Language Understanding Evaluation (GLUE)* benchmark (Wang et al., 2018) is a collection of diverse NLP datasets to evaluate the performance of models on a wide range of tasks, including natural language inference (NLI), semantic textual similarity (STS), and sentiment analysis. Some of the datasets included in GLUE are the Stanford Sentiment Treebank (SST-2) (Socher et al., 2013), the Multi-Genre NLI Corpus (MNLI) (Williams et al., 2018), and the Corpus of Linguistic Acceptability (Warstadt et al., 2019). The GLUE benchmark also includes a leaderboard that ranks the performance of different models on the tasks in the benchmark as well as the human performance on the tasks. However, shortly after the release of GLUE, many NLP models started to achieve superhuman performance and a new, more challenging benchmark called *SuperGLUE* (Wang et al., 2019) was introduced to address this issue. SuperGLUE includes more difficult tasks and datasets compared to GLUE, such as Winograd Schema Challenge (WSC) (Levesque et al., 2012), Reading Comprehension with Commonsense Reasoning (ReCoRD) (Zhang et al., 2018), Choice of Plausible Alternatives (COPA) (Gordon et al., 2012), and Word-in-Context (WiC) (Pilehvar and Camacho-Collados, 2019). Despite the increased difficulty of the tasks in SuperGLUE, many models have achieved high performance on the benchmark that exceeded the human performance, demonstrating both the rapid progress in NLP research, and the limitations of these benchmarks in capturing the full range of human language understanding capabilities.

BIG-Bench The *BIG-Bench* benchmark (**?**) is a large-scale evaluation benchmark, that contains 204 tasks, contributed by 450 authors across 132 institutions. The tasks in Big-Bench are designed to be more challenging and diverse compared to existing benchmarks, covering a wide range of NLP tasks, including text classification, question answering, summarization, dialogue generation and many more. Despite the inclusion of only specifically challenging tasks, several of the recent state-of-the-art models are shown to surpass average human performance and close to expert human performance on the benchmark.

MMLU The *Measuring Massive Multitask Language Understanding (MMLU)* benchmark (Hendrycks et al., 2021) is a multiple-choice question answering benchmark that covers 57 tasks including elementary mathematics, US history, computer science, law and more. Achieving a high score on this benchmark requires a strong world knowledge as well as problem-solving ability, making it a challenging benchmark for the models. As of the writing of this dissertation, MMLU remains to be one of the most popular and challanging benchmarks for evaluating the large language models.

Chatbot Arena The introduction of ChatGPT and other large language models has led to a surge in public interest and interaction with these AI systems. Whereas previously these models were primarily used by researchers, the web-based interfaces provided by the companies that developed them have made them accessible to a much broader audience. The tasks that users now ask these models to address are incredibly diverse and ever-changing. This presents a challenge for traditional static benchmarks, which can only capture a small portion of the wide range of real-world queries and problems that humans pose to the models. Even the most comprehensive benchmark will inevitably fall short of encompassing the full scope of how these models are being utilized in practice. To address this issue, *Chatbot Arena* was introduced (Zheng et al., 2023), a crowdsourced battle platform, where users get responses to their queries from multiple hidden models and vote on the best response. Since the models compete head-to-head, their performance is measured using ELO ratings, which are commonly used in chess to rank players based on their performance against each other. Due to its dynamic nature and coverage of virtually unlimited types of real world queries and tasks, Chatbot Arena became a popular platform to track the progress of the large language models and to evaluate their performance in real-world scenarios.

1.4 Future Work

Building on the work presented in this thesis, several promising avenues for future research have emerged. These directions not only aim to extend the current findings but also address some of the unresolved challenges in the evaluation of language models and the enhancement of NLP capabilities for low-resource languages.

Exploiting Multi-Parallel Corpora The use of multi-parallel corpora still presents untapped potential for improving multilingual NLP tasks. In this work, we showed that leveraging multi-parallel corpora can improve word alignment quality. Future research could focus on developing more sophisticated algorithms that can leverage the rich linguistic information available in multi-parallel corpora to enhance the performance of multilingual language models, especially for low-resource languages.

New Evaluation Benchmarks The development of more comprehensive and challenging evaluation benchmarks remains a critical need. As our models become more sophisticated and capable, they saturate existing benchmarks quickly, making it challenging to assess their true capabilities. Moreover, despite their impressive performance, it is shown that the current state-of-the-art models still break and fail in unexpected ways. It is important to create datasets that can reveal such limitations and vulnerabilities of the models in order to ensure their safe and reliable deployment in real-world applications.

Another important direction is creating more evaluation benchmarks for low-resource languages. The current benchmarks are primarily focused on high-resource languages, which limits the evaluation of models on languages with limited resources. Developing benchmarks that are specifically tailored for low-resource languages can help assess the performance of models on these languages and guide the development of more effective methods for enhancing their capabilities.

Enhanced Transfer Learning for Low-Resource Languages Despite the progress made in transferring knowledge from high-resource to low-resource languages, significant challenges remain due to linguistic diversity and resource disparities. In this work, we explored the effectiveness of leveraging a high resource cousin to improve the multilingual model performance on low-resource languages in a transfer learning setting. Several other studies also show that joint training of multiple languages can improve the performance of language models on low-resource languages if they have high resource cousins in the training data (ImaniGooghari et al., 2023). A more systematic and detailed investigation of the impact of the linguistic similarity between languages on the transfer learning performance can provide valuable insights into how to effectively leverage high-resource languages to enhance the capabilities of models on low-resource languages.

Chapter 2

Kardeş-NLU: Transfer to Low-Resource Languages with Big Brother's Help – A Benchmark and Evaluation for Turkic Languages

Kardeş-NLU: Transfer to Low-Resource Languages with Big Brother's Help – A Benchmark and Evaluation for Turkic Languages

Lütfi Kerem Şenel[1,2,*], **Benedikt Ebing**[3,*], **Konul Baghirova**[1]
Hinrich Schütze[1,2] **and Goran Glavaš**[3]
[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2]Munich Center for Machine Learning (MCML), Germany
[3]University of Würzburg
lksenel@cis.lmu.de,
{benedikt.ebing, goran.glavas}@uni-wuerzburg.de

Abstract

Cross-lingual transfer (XLT) driven by massively multilingual language models (mmLMs) has been shown largely ineffective for low-resource (LR) target languages with little (or no) representation in mmLM's pretraining, especially if they are linguistically distant from the high-resource (HR) source language. Much of the recent focus in XLT research has been dedicated to *LR language families*, i.e., families without any HR languages (e.g., families of African languages or indigenous languages of the Americas). In this work, in contrast, we investigate a configuration that is arguably of practical relevance for more of the world's languages: XLT to LR languages that do have a close HR relative. To explore the extent to which a HR language can facilitate transfer to its LR relatives, we (1) introduce Kardeş-NLU,[1] an evaluation benchmark with language understanding datasets in five LR Turkic languages: Azerbaijani, Kazakh, Kyrgyz, Uzbek, and Uyghur; and (2) investigate (a) intermediate training and (b) fine-tuning strategies that leverage Turkish in XLT to these target languages. Our experimental results show that both—integrating Turkish in intermediate training and in downstream fine-tuning—yield substantial improvements in XLT to LR Turkic languages. Finally, we benchmark cutting-edge instruction-tuned large language models on Kardeş-NLU, showing that their performance is highly task- and language-dependent.

1 Introduction

Transformer-based massively multilingual language models (mmLMs), such as mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020a), and mT5 (Xue et al., 2021), have substantially advanced multilingual NLP. These models have enabled rapid development of language technologies for a wide range of low-resource (LR) languages by means of cross-lingual transfer (XLT) from high-resource (HR) languages, using zero-shot (Wu and Dredze, 2019; Karthikeyan et al., 2020) or few-shot transfer techniques (Lauscher et al., 2020; Schmidt et al., 2022). mmLMs are, however, biased towards HR languages and XLT with mmLMs yields especially poor transfer performance for LR target languages that are (i) underrepresented in mmLMs' pretraining corpora and (ii) linguistically distant from the source language (Lauscher et al., 2020). Besides these reasons, such poor XLT is also a consequence of the *curse of multilinguality* (Conneau et al., 2020a; Pfeiffer et al., 2022), i.e., a reduced representational quality of supported languages, stemming from mmLMs' parameters being shared by many linguistically diverse languages.

In recent years, a large body of work focused on improving XLT abilities of mmLMs, ranging from models that aim to better align representation subspaces of source and target language with cross-lingual supervision (Cao et al., 2020; Hu et al., 2021; Conneau et al., 2020b; Minixhofer et al., 2022; Wang et al., 2022) to those that improve the mmLMs' representational capacity for individual, mostly LR languages (Pfeiffer et al., 2020; Parović et al., 2022; Ansell et al., 2021; Pfeiffer et al., 2022). At the same time, an incredible amount of effort has also been dedicated to the creation of new multilingual evaluation benchmarks that either encompass sets of linguistically diverse languages (Clark et al., 2020; Ponti et al., 2020; Ruder et al., 2021) or focus on LR languages (Adelani et al., 2021; Muhammad et al., 2022; Ebrahimi et al., 2022; Armstrong et al., 2022; Winata et al., 2023; Khanuja et al., 2023, *inter alia*). The vast majority of existing work, however, assumes (i) zero-shot downstream transfer from (ii) English as the source. That is primarily because, on the one hand, for most tasks, training data is only available in English. On the

[*] These authors contributed equally.
[1] https://github.com/lksenel/Kardes-NLU

other hand, many of the recent benchmarks cover *LR language families*, i.e., families without *any* HR languages (e.g., some African language families or indigenous languages of the Americas): this prevents the creation of high-quality silver-standard training data in a (closely) related HR language (e.g., via machine translation (MT)), as no such language exists.

Contributions. 1) In this work, we contribute to the body of evaluation resources for LR XLT with Kardeş-NLU,[2] an evaluation benchmark covering three natural language understanding (NLU) tasks—natural language inference (NLI), semantic text similarity (STS), and commonsense reasoning, in particular choice of plausible alternatives (COPA)—for five Turkic languages—Azerbaijani (az), Kazakh (kk), Kyrgyz (ky), Uyghur (ug), and Uzbek (uz). We focus on Turkic languages because, unlike most concurrent work, we aim to explore a highly underinvestigated XLT research question: to what extent can LR languages that *do have* a linguistically and genealogically (close) HR relatives profit from those relatives (Snæbjarnarson et al., 2023). **2)** We extend a number of established (i) intermediate training and (ii) fine-tuning approaches (covering both zero-shot and few-shot XLT) for improving LR XLT by incorporating Turkish as the HR sibling of the Kardeş-NLU languages; and show that the mixture of incorporating Turkish in intermediate training and in task-specific fine-tuning results in substantial performance gains. **3)** Given the praised generalization abilities of large instruction-based language models (LLMs) (Chung et al., 2022; Ahuja et al., 2023; Asai et al., 2023), we additionally evaluate (zero-shot) two multilingual LLMs on Kardeş-NLU—the open mT0 (Muennighoff et al., 2023) and commercial ChatGPT—showing that their performance is highly task- and language-dependent and in some cases substantially trails that of XLT with traditionally fine-tuned "small" mmLMs.

2 Kardeş-NLU Benchmark

Language and Task Selection. We selected languages for Kardeş-NLU based on two criteria: (i) linguistic and genealogical diversity within the Turkic language family and (ii) availability of native speakers of those languages who are also fluent in English.[3] Our final selection contains five languages from the Common Turkic branch, covering three different sub-branches: Western Oghuz languages (Azerbaijani; Turkish, as the HR language in our experiments, also belongs to this branch), Kipchak languages (Kazakh and Kyrgyz) and Karluk languages (Uzbek and Uyghur). Moreover, Kardeş-NLU covers languages with two different scripts: Latin (Azerbaijani and Uzbek) and Cyrillic (Kazakh, Kyrgyz, and Uyghur).[4]

We select three tasks that are (i) among the most prominent NLU tasks, included in popular NLU benchmarks (Wang et al., 2018, 2019), and (ii) already have existing evaluation datasets in a number of languages (commonly translations of an original English dataset): NLI (Conneau et al., 2018; Aggarwal et al., 2022; Ebrahimi et al., 2022), STS (Cer et al., 2017), and COPA (Gordon et al., 2012; Ponti et al., 2020).

Dataset Translation. We adopt a widely used two-step translation approach to obtain translations in which a native speaker of the target language, fluent in English, post-edits the output of MT.[5] This way, we translated English instances from the following datasets: XNLI (Conneau et al., 2018) (2000 instances from the test portion and 1000 instances from the validation portion), STS-Benchmark (Cer et al., 2017) (800 test instances and 200 validation instances), and XCOPA (Ponti et al., 2020) (500 test instances and 100 validation instances). We initially manually compared, on a small subsample of instances from all three datasets, translation (i) with Google Translate (GT) vs. the open Turkic Interlingua MT models (Mirzakhalov et al., 2021) and (ii) from English vs. from Turkish (with Turkish instances that were, in turn, machine translated from English) and have found that GT from English produces the best output. Due to MT in the first step, we instructed the annotators to pay special attention to the idiomaticity of the source English sentences during post-editing. This particularly refers to finding suitable translations for culture-specific concepts that do not have a direct translation (e.g.,

[2] *kardeş* is a Turkish gender-neutral word for *sibling*. Referring to a brother (*erkek kardeş*) or sister (*kız kardeş*), requires an additional gender denotation: *kız* (*girl*) or *erkek* (*boy*).

[3] For example, we wanted to include Chuvash, the only living language of the Oghur branch of Turkic languages, but we could not find annotators native in that language.

[4] While Uyghur is more commonly written in the Arabic script (e.g., in CC-100 or Wikipedia), our Uyghur annotator was unfamiliar with it and was only able to produce Uyghur translations in the Cyrillic script.

[5] We hired one annotator per target language.

"passing for white" has no direct translation in our target languages since *racial passing* is not a native concept in respective cultures). Table 1 displays several instances from Kardeş-NLU.

Annotation Costs. Given the high post-editing costs, Kardeş-NLU contains only subsets of the original English development and test portions of STS-B and XNLI. All of our annotators were university students who were paid the equivalent of 14$ per hour for their effort. On average, post-editing took 92 hours per language, bringing the total cost of creating Kardeş-NLU to 6,440$.

3 Kardeş Transfer: Leveraging Turkish

We next attempt to improve XLT to LR Kardeş-NLU languages by explicitly incorporating Turkish as the close HR relative into the process. We try to (1) increase mmLMs' capacity for the target languages as well as their alignment with Turkish via intermediate LM training and (2) leverage Turkish as an additional source language in downstream zero-shot and few-shot transfer.

3.1 Intermediate Language Modeling

Adapting pretrained mmLMs to target distributions—different languages, domains, or datasets—through further LM-ing can bring significant performance gains (Howard and Ruder, 2018; Gururangan et al., 2020; Muller et al., 2021; Wang et al., 2022; Hung et al., 2022). Building upon these findings, we investigate the benefit of additional LM-ing in transfer to LR Kardeş-NLU languages. Specifically, we explore the potential benefits of incorporating Turkish into the mmLM adaptation process and the extent to which this inclusion can improve the downstream performance for LR Turkic languages. We experiment with three different intermediate training strategies detailed below: in all cases, we (1) use the standard masked language modeling (MLM) as the training objective and (2) update all of the mmLM's pretrained weights.

Target Language LM-ing (TLLM). In this case, we perform additional MLM-ing only on the limited-size corpora of the target language. Turkish, as the HR relative, is not leveraged in TLLM.

Bilingual Alternating LM-ing (BALM). Here we alternately update the mmLM by MLM-ing on one batch of target language data, followed by one batch of Turkish data. BALM is similar to the bilingual training procedure of Parović et al. (2022): they, however, opt for parameter-efficient training with adapters, whereas we update all of the mmLM's parameters.

Bilingual Joint LM-ing (BJLM). Like BALM, in BJLM we perform bilingual MLM-ing on both the LR target language and the related HR language (Turkish). However, while in BALM monolingual batches are alternated, in BJLM batches are bilingual, i.e., they consist of instances of both languages. Importantly, both languages have the same number of instances in each batch (i.e., B/2 with B as the batch size). Although such balancing leads to frequent repetition of instances from the LR language corpus, these repeating instances are, in different batches, "regularized" with different source-language instances, which prevents overfitting to small-sized corpora of LR languages. Schmidt et al. (2022) demonstrate the effectiveness of BJLM in task-specific few-shot fine-tuning; here, we test it in intermediate MLM-ing.

Parameter-Efficient LM-ing. Besides full fine-tuning, we also carried out intermediate training (for TLLM and BALM) in a parameter-efficient manner with adapters (Houlsby et al., 2019) in the vein of prior work on XLT (Pfeiffer et al., 2020; Parović et al., 2022). Adapter-based variants yielded consistently weaker performance compared to tuning all mmLM's parameters. For brevity, we report these results in the Appendix (§C).

3.2 Downstream Cross-Lingual Transfer

We investigate two common setups for downstream cross-lingual transfer: (1) zero-shot XLT, in which we assume that we do not have any labeled task instances in the target language, and (2) few-shot transfer, in which a small number of labeled instances in the target language exists. We follow the fair XLT evaluation procedure of Schmidt et al. (2022), which does not allow for model selection based on target-language validation data. Relying on target-language validation violates the assumption of true zero-shot XLT. Moreover, Schmidt et al. (2022, 2023a) show that any labeled target-language instances are better leveraged for training. We thus use the validation portions of Kardeş-NLU

Language	Task	Instance	Label
Azerbaijani	NLI	*Premise*: Bütün hallarda müştərinin iddialarına xələl gətirməmək üçün mühüm addımlar atılmalıdır. (*In all cases, significant steps would have to be taken to avoid prejudicing the client's claims.*) *Hypothesis*: Bu addımlara müştərilərin həqiqi şəxsiyyətinin müstəntiqlərdən gizlədilməsi daxildir (*These steps include hiding the real identity of clients from investigators.*)	Neutral
Kazakh	STS	*Sent. 1*: Бір адам қазанға күріш слаып жатыр. (*A man pours rice into a pot.*) *Sent. 2*: Ер адам табаққа күріш салып жатыр. (*A man is putting rice in a bowling pot.*)	4.2
Kyrgyz	COPA	*Premise*: Кыз кодду жаттап калды. (*The girl memorized the code.*) *Choice 1* (Cause): Ал өзүнө өзү окуду. (*She recited it to herself.*) *Choice 2* (Cause): Ал муну жазууну унутуп калды. (*She forgot to write it down.*)	Choice 1
Uzbek	STS	*Sent. 1*: Okapi daraxtdan yemoqda. (*An okapi is eating from a tree.*) *Sent. 2*: Sichqon suv purkagichdan ichadi. (*A moose drinks from a sprinkler.*)	0.3
Uyghur	COPA	*Premise*: Дәрәх йопурмақлирини төкти. (*The tree shed its leaves.*) *Choice 1* (Effect): Йопурмақ рәңгигә боялди. (*The leaves turned colors.*) *Choice 2* (Effect): Йопурмақлар йәргә йиғилип қалди. (*The leaves accumulated on the ground.*)	Choice 2

Table 1: Examples from Kardeş-NLU one for each language and at least one for each task.

only for training in few-shot XLT.

Zero-Shot Transfer. We explore three zero-shot XLT setups: (i) monolingual training on English data, (ii) monolingual training on Turkish data, machine translated from the original English training data, and (iii) bilingual training on both English and machine-translated Turkish data, with joint bilingual batches.

Few-Shot Transfer. In few-shot fine-tuning, we additionally train on a small number of instances in the target language. We evaluate two different few-shot fine-tuning strategies: (1) in *sequential* transfer (Lauscher et al., 2020; Zhao et al., 2021), large(r)-scale fine-tuning on data from the source language(s)—in our case, English, Turkish, or bilingually English and Turkish—is followed by efficient target-language fine-tuning on the few shots; (2) in *joint* fine-tuning, we follow Schmidt et al. (2022) and, after initial source-only training, interleave source- and target-language instances at the batch level—the final batch loss is then the macro-average of the language-specific losses. Note that this results in joint trilingual fine-tuning when the source datasets are both English and Turkish.

	az	kk	ky	ug	uz
script	Latin	Cyrillic	Cyrillic	Arabic	Latin
monolingual corpus sizes (in bytes)					
CC-100	1.3G	889M	173M	46M	155M
Wiki	315M	354M	126M	36M	136M
Avg no. tokens in test instances (XLM-R tokenizer)					
NLI	44	46	47	79	52
COPA	22	24	24	34	26
STS	34	36	36	56	40

Table 2: Dataset statistics for Wikipedias and CC-100 portions of Kardeş-NLU languages along with average no. tokens in the test instances of Kardeş-NLU (as per XLM-R tokenizer)

4 Experimental Setup

Data. We carry out intermediate training for five Kardeş-NLU languages, monolingually (i.e., TLLM) or bilingually with Turkish (BALM and BAJM, see §3.1) using Wikipedias of the respective languages. Table 2 summarizes the base statistics of Wikipedias of Kardeş-NLU languages,[6] together with the size of their corresponding monolingual corpora in CC-100.[7] The sizes of the Turkish Wikipedia and Turkish CC-100 portions are 631MB and 5.4GB, respectively. Table 2 additionally shows the average number of tokens in test instances after XLM-R tokenization. Uyghur yields substantially more tokens than the other four languages. This is because most of Uyghur's pretraining corpus in XLM-R's is in the Arabic script, whereas Uyghur instances in Kardeş-NLU are written in Cyrillic.

In downstream XLT, we use the existing training data in English and respective automatic translations to Turkish. For NLI, we train on MNLI (Williams et al., 2018) and (automatically translated) Turkish training data from XNLI (Conneau et al., 2018). For STS, we train on the English training portions of STS-B (Cer et al., 2017) and its existing (automatic) translation to Turkish.[8] Due to

[6] The Wikipedia dumps were obtained from https://dumps.wikimedia.org/ on 10.12.2022. The text is extracted using the standard wikiextractor script.

[7] We report CC-100 portions, as XLM-R—the mmLM that we use in our experiments—was pretrained on it.

[8] https://huggingface.co/datasets/emrecan/stsb-mt-turkish

the small size of the English training data for COPA (400 instances) (Gordon et al., 2012), reported to hinder convergence of mmLM-based models (Sap et al., 2019; Ponti et al., 2020), we follow this prior work and first fine-tune on (English) SocialIQa (SIQA)—a closely related causal commonsense reasoning dataset (Sap et al., 2019) before fine-tuning on (English and/or Turkish) COPA data[9].

Intermediate Training Details. In all our main experiments, we use XLM-R (Base size) (Conneau et al., 2020a) as our mmLM. For the bilingual intermediate training procedure (e.g., BALM and BJLM), we train for a full epoch on Turkish Wikipedia: this results in multiple passes over the target language Wikipedias, given that those are substantially smaller. Thus, in the interest of fair evaluation, we train TLLM for multiple epochs: 2 for Azerbaijani and Kazakh, 5 for Kyrgyz and Uzbek, and 18 for Uyghur. We set the batch size to 32 and limit the sequence length to 128 tokens. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a fixed learning rate of $5e{-}5$.

Downstream Training Details. We adopt standard fine-tuning and add a task-specific classifier on top of the mmLM. Unless explicitly said otherwise, we perform full fine-tuning updating all parameters of the encoder together with the classifier's parameters. For NLI and STS, we encode the pair of sentences with the mmLM and feed the transformed representation of the [CLS] token to the classifier. For the multiple-choice tasks—COPA and SIQA (which we use as a "pre-fine-tuning" task to stabilize COPA training)—we face a varying number of answer choices per dataset (i.e., there are 3 possible answers in SIQA and 2 in COPA). We follow prior work Sap et al. 2019; Ponti et al. 2020 and encode the premise together with each answer choice. We feed the resulting output [CLS] token into a feed-forward regressor that produces a single score for each answer choice. Afterwards, the individual scores of all choices are concatenated and fed to the softmax classifier.

We train the models for 10 epochs with mixed precision using AdamW (Loshchilov and Hutter, 2019) with a weight decay of 0.05 and the initial learning rate set to $2e{-}5$. We use a linear scheduler with 10% linear warm-up and decay. We deviate from this configuration (i) in the *joint* few-shot fine-tuning, where we train for 50 epochs without a scheduler, following recommendations of (Schmidt et al., 2022), and (ii) for all NLI experiments, where we train for 5 epochs due to the size of the MNLI training data (ca. 400K instances). The sequence length is limited to 128 tokens for all tasks, matching the input size of the intermediate MLM-ing. We fine-tune with a batch size of 32, except in the trilingual *joint* few-shot fine-tuning (English-Turkish-target language), where we sample 10 instances per language (i.e., batch size 30). For each experiment, we execute three runs with different random seeds and report the average performance (accuracy for NLI and COPA and Pearson correlation for STS). In zero-shot XLT, we report the performance of the last checkpoint obtained at the end of the training. In few-shot XLT, we start training from the last snapshot of the source training (English, Turkish, or English and Turkish) and select the last snapshot of the second—*sequential* or *joint*—training step.

5 Results and Discussion

Zero-Shot Transfer. Table 3 displays the zero-shot XLT performance for all five Kardeş-NLU languages on NLI, COPA and STS. Generally, we reach the best performance when Turkish is integrated into both intermediate training (rows BALM and BAJM) *and* as the source language in fine-tuning (columns TR and EN,TR). On average, across all five languages, BJLM combined with source fine-tuning on concatenated English and Turkish instances (EN,TR) yields a 6.6% and 2.1% boost over zero-shot XLT from English only with the vanilla XLM-R (Base) on NLI and COPA, respectively. On these two tasks, this observation holds for all individual languages except Kazakh. The gains over the vanilla zero-shot XLT for STS, however, are much smaller, with only BALM combined with English and Turkish fine-tuning surpassing the default zero-shot XLT performance of XLM-R (Base, EN) and that by a narrower margin (+0.6). We speculate that this is because (i) fine-grained sentence similarity is more sensitive to slight semantic misalignment and (ii) while our bilingual intermediate training improves the semantic links between Turkish and the target language, it is not of an adequate scale to establish alignments of such semantic precision.

Including Turkish as a fine-tuning source language (TR and EN,TR) brings consistent gains

[9] We translate the COPA training set to Turkish with GT.

over transfer from English only, regardless of the intermediate training strategy. The best results are almost always obtained when we fine-tune on both English and Turkish (EN,TR): we hypothesize that such fine-tuning establishes task-specific representational associations between the two languages and allows the transfer to benefit from both (i) XLM-R's unmatched representational quality for English and (ii) proximity of Turkish to the target languages. The effect is then further amplified when intermediate training (BALM and BJLM) increases the XLM-R's capacity for Turkish and the target language and strengthens the alignments between them. This is confirmed by the fact that intermediate training on the target language alone (TLLM) brings downstream gains (compared to Base) for NLI but not for the other two tasks.

Looking at individual languages, we observe the least (and smallest) gains for Azerbaijani and Kazakh, the two most-resourced Kardeş-NLU languages, and the most (and largest) gains for the three less-resourced languages: Uyghur, Uzbek, and Kyrgyz (e.g., compared to Base transfer from EN on NLI, BJLM with transfer from EN,TR leads to gains of 5.0% for Kyrgyz, 5.1% for Uzbek, and 17.2% for Uyghur). We see the largest gains (by a wide margin) for Uyghur, despite the script mismatch between the intermediate training (Arabic script) and evaluation (Uyghur in Cyrillic script). The intermediate bilingual training for Uyghur, which improves representations of Arabic-script tokens, would thus likely yield even larger gains if the Uyghur test instances were in the Arabic script.

Few-Shot Transfer. Table 4 summarizes the few-shot XLT results. We observe mixed results compared to the strongest zero-shot approaches: while there is a small improvement on STS (+1.0%), we see virtually no gains for COPA (+0.1%) and NLI (-0.3%). Consistent with zero-shot XLT findings, few-shot XLT yields best results when we start the few-shot target language training from models trained on both English and Turkish (EN,TR). Additionally, we observe that few-shot XLT with models that were intermediately trained on Turkish and the target languages (BALM, BAJM) yields stronger performance than with those MLM-ed on the target language alone (TLLM). Nonetheless, there is no bilingual intermediate training strategy that is consistently best: BJLM yields better scores on COPA, whereas BALM reaches better STS performance; on NLI, both strategies perform comparably. Concerning the number of target language shots, we observe that we typically need at least 50 shots to match or surpass the zero-shot XLT performance. Comparing few-shot transfer procedures, we observe task-dependent variability. On NLI, sequential fine-tuning substantially outperforms the joint approach. Conversely, on COPA and STS, joint few-shot transfer shows better performance, with a more pronounced gap on STS.

Kardeş-NLU: A Difficult Few-Shot XLT Benchmark. Not only does the comparison of zero-shot and few-shot results in Table 4 render Kardeş-NLU as a difficult few-shot XLT benchmark but also does Kardeş-NLU involve two tasks—STS and COPA—that are underrepresented in the current body of work on (few-shot) XLT (Lauscher et al., 2020; Zhao et al., 2021; Schmidt et al., 2022). This makes Kardeş-NLU a valuable evaluation resource for XLT research.

Instruction-Based LLMs on Kardeş-NLU. Given the recent popularity of instruction-tuned LLMs as competent "generalizers" (Ouyang et al., 2022; Ahuja et al., 2023), we additionally evaluate (zero-shot) two state-of-the-art multilingual LLMs on Kardeş-NLU:[10] mT0 (Muennighoff et al., 2023), as the open model tuned on instructions derived from NLP tasks, and ChatGPT, as the commercial model tuned from human instructions and feedback. To this end, we slightly modify the instructions and prompts proposed by Ahuja et al. (2023): we provide further details in the Appendix §A.

Figure 1 compares the best zero-shot XLT performance (based on XLM-R) for each language from Table 3 against zero-shot inference with mT0 and ChatGPT. The NLI results, in which both LLMs dramatically underperform our language-adapted zero-shot XLT (-23.9% and -15.1% for ChatGPT and mT0, respectively), diametrically oppose those on COPA, where both LLMs (and especially mT0) excel and surpass our best zero-shot XLT (the gap is full 10% in favor of mT0, albeit only 1.1% for ChatGPT). We believe that this is because mT0 was instruction-tuned, multilingually, on a large number of different multi-choice QA datasets (including, e.g., SIQA). ChatGPT, in contrast, being fine-tuned based on open-ended instruction-reply

[10] Regression (i.e., score prediction) tasks are inherently difficult to cast as text generation tasks; we thus omit STS from this evaluation.

		Azerbaijani			Kazakh			Kyrgyz			Uyghur			Uzbek			Average		
		EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR
NLI	Base	76.5	**80.1**	79.6	73.8	76.3	**77.3**	70.4	73.9	74.1	42.2	44.4	42.9	70.7	72.0	71.8	66.7	69.4	69.1
	TLLM	77.3	79.0	79.2	75.3	76.3	76.8	72.4	74.1	74.4	56.7	57.1	56.9	73.1	74.3	74.8	71.0	72.2	72.4
	BALM	77.3	78.8	79.3	74.4	75.3	77.0	71.6	73.4	74.0	57.4	58.7	58.0	73.1	74.5	75.0	70.8	72.1	72.7
	BJLM	76.4	78.4	79.3	74.9	75.1	76.8	71.9	74.3	**75.5**	57.2	59.2	**59.4**	73.4	74.6	**75.7**	70.7	72.3	**73.3**
COPA	Base	60.1	61.1	60.9	60.7	**60.8**	59.9	59.7	60.0	59.4	51.8	52.7	52.7	57.3	59.5	60.1	57.9	58.8	58.6
	TLLM	62.1	62.1	61.5	55.7	55.8	56.1	57.5	59.7	58.9	49.9	50.3	49.3	62.9	**63.2**	62.5	57.6	58.2	57.7
	BALM	57.2	58.3	59.4	59.1	59.5	59.7	56.1	59.9	59.1	51.1	**53.9**	52.5	60.5	61.7	61.9	56.8	58.6	58.5
	BJLM	61.8	**63.3**	**63.3**	58.4	58.6	57.7	56.8	61.5	**62.0**	50.9	52.2	**53.9**	61.7	60.5	62.9	57.9	59.2	**60.0**
STS	Base	80.3	78.9	**80.4**	**85.8**	84.1	84.8	78.2	77.9	**78.7**	69.2	64.8	64.2	78.3	77.2	77.1	78.4	76.6	77.1
	TLLM	75.8	75.5	78.1	80.6	80.1	81.9	71.3	71.8	74.2	70.6	69.3	71.3	70.6	67.0	76.9	73.8	72.7	76.5
	BALM	72.7	78.7	79.7	81.4	83.2	83.9	71.1	77.3	78.3	72.8	72.3	**73.5**	72.5	77.6	**79.3**	74.1	77.8	**79.0**
	BJLM	69.3	77.0	78.3	78.6	83.2	84.6	69.9	75.1	77.3	65.7	66.9	69.0	71.1	76.8	77.3	70.9	75.8	77.3

Table 3: Zero-Shot XLT results on Kardeş-NLU for three intermediate LM-ing strategies (TLLM, BALM, and BJLM) and source fine-tuning datasets (English only, Turkish only, and English and Turkish combined). The best results for each language-task pair are shown in **bold**. The evaluation metrics are accuracy (%) for NLI and COPA, and Pearson correlation for STS.

		Zero-Shot			Few-Shot																	
					Sequential									Joint								
		EN	TR	EN,TR	EN			TR			EN,TR			EN			TR			EN,TR		
Shots		-	-	-	10	50	100	10	50	100	10	50	100	10	50	100	10	50	100	10	50	100
NLI	Base	66.7	69.4	69.1	63.5	67.9	68.1	65.7	69.0	69.3	66.0	69.5	70.1	65.0	66.2	66.4	67.0	67.4	67.5	66.7	68.0	69.0
	TLLM	71.0	72.2	72.4	68.1	70.7	71.7	69.3	71.9	72.3	70.6	72.6	72.5	69.3	70.3	70.7	70.1	71.3	70.7	70.4	71.2	71.9
	BALM	70.8	72.1	72.7	67.9	70.9	71.2	69.0	71.8	72.0	70.0	72.6	73.0	69.1	70.0	70.4	70.5	71.5	71.3	70.5	71.0	71.6
	BJLM	70.7	72.3	**73.3**	67.5	71.0	71.3	69.2	71.7	71.5	69.9	72.7	73.0	69.4	70.3	69.9	70.7	71.3	71.2	70.6	71.5	71.8
COPA	Base	57.9	58.8	58.6	56.4	57.9	58.8	56.8	57.6	58.2	57.0	57.8	58.3	57.6	57.9	59.0	58.7	58.5	58.5	59.0	59.0	59.5
	TLLM	57.6	58.2	57.7	56.8	57.4	58.4	57.1	57.9	59.5	56.7	58.0	58.9	57.2	57.5	58.3	58.1	58.7	58.6	58.6	59.0	59.8
	BALM	56.8	58.6	58.5	56.6	57.2	58.1	56.8	58.0	58.5	57.6	58.0	58.4	56.8	57.8	57.2	59.0	58.7	58.2	59.1	59.4	58.3
	BJLM	57.9	59.2	**60.0**	57.2	58.6	59.3	58.0	59.3	59.7	58.0	59.8	59.8	58.1	58.8	58.8	58.9	59.9	59.3	60.1	59.9	59.8
STS	Base	78.4	76.6	77.1	73.5	75.5	75.4	74.5	76.5	75.7	75.4	77.1	77.1	76.3	77.6	77.6	77.0	78.9	78.9	77.1	79.0	79.3
	TLLM	73.8	72.7	76.5	73.6	75.3	75.6	74.9	76.1	76.2	76.4	77.3	77.6	75.1	76.8	76.9	75.2	77.0	77.6	77.2	78.5	78.8
	BALM	74.1	77.8	**79.0**	74.5	76.0	76.3	76.2	77.6	77.8	77.3	78.6	78.4	77.1	77.2	76.9	78.3	79.4	79.6	79.4	80.0	80.0
	BJLM	70.9	75.8	77.3	72.8	74.9	75.4	75.2	76.9	76.8	76.1	77.7	78.1	74.0	76.2	76.6	76.8	78.3	78.5	77.9	79.3	79.4

Table 4: Results of *sequential* and *joint* few-shot XLT on Kardeş-NLU: performance with 10, 50, and 100 target-language shots. The best zero-shot result per task is shown in **bold**, the best few-shot result is underlined. The evaluation metrics are accuracy (%) for NLI and COPA, and Pearson correlation for STS.

pairs, has a weaker inductive bias for both COPA and NLI. The two LLMs yield the best performance on both tasks for Azerbaijani, the most resourced language in Kardeş-NLU—the performance drops for the remaining languages are drastic, especially for ChatGPT. This is in line with findings from concurrent work (Ahuja et al., 2023; Asai et al., 2023) and shows that even the largest instruction-tuned LLMs are bound by the language distribution of their (pre)training data, indicating that there is still a long way to go to enable truly multilingual NLP.

6 Related Work

Multilingual Evaluation Benchmarks. Reliable evaluation of the multilingual abilities of mmLMs requires that they are tested against a large set of diverse languages (Joshi et al., 2020). On the one hand, multilingual benchmarks that encompass many tasks, such as XGLUE (Liang et al., 2020) and XTREME (Hu et al., 2020; Ruder et al., 2021), comprise diverse but predominantly highly or moderately resourced languages: their coverage of LR languages is small and varies across tasks. On the other hand, many recent efforts introduce dedicated benchmarks for specific families of LR languages (Armstrong et al., 2022; Adelani et al., 2022; Ebrahimi et al., 2022; Winata et al., 2023, *inter alia*). While these target truly underrepresented languages, they typically focus on a single task only, e.g., NLI or NER. With Kardeş-NLU we, (i) cover multiple languages from an underrepresented language family while (ii) including various tasks (NLI, COPA, and STS) that require different

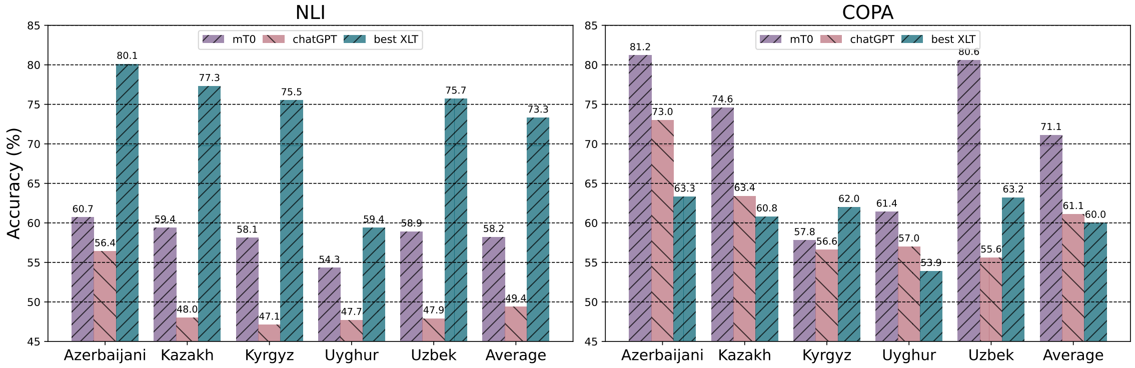


Figure 1: Performance of mT0-XXL, chatGPT, and our best performing zero-shot XLT strategy on NLI and COPA.

degrees of precision in language understanding.

Cross-Lingual Transfer with mmLMs. mmLMs still play an important role in multilingual NLU and XLT, exhibiting good performance in zero-shot XLT (Wu and Dredze, 2019; Hu et al., 2020) to HR languages. They, however, perform much worse in XLT to LR languages distant from English (as the common source). The body of work on improving XLT is threefold. The first line of work seeks to improve XLT via post-hoc alignment of representational subspaces of individual languages, guided by parallel data (Cao et al., 2020; Conneau et al., 2020b; Hu et al., 2021; Wang et al., 2022; Minixhofer et al., 2022, *inter alia*) and driven by cross-lingual supervision. These efforts, however, offer little gain for LR languages, whose representational subspaces are of low semantic quality, to begin with. The second line of work seeks to improve the representational quality for LR languages through additional language modeling training (Pfeiffer et al., 2020; Ansell et al., 2021; Parović et al., 2022; Pfeiffer et al., 2022), resulting in moderate downstream performance gains. Finally, the third line of work (Lauscher et al., 2020; Zhao et al., 2021; Xu and Murray, 2022; Schmidt et al., 2022, 2023a,b) focuses on the actual downstream transfer, rather than the task-agnostic adaptation of mmLMs, investigating how to best utilize the limited number of annotated task-specific target-language instances (Lauscher et al., 2020; Schmidt et al., 2022, 2023a) or tailor source-language instances to resemble target language ones (Xu and Murray, 2022).

In this work, we adopt the latter two ideas and seek to improve XLT to Turkic LR languages via both intermediate LM-ing and few-shot XLT: unlike most existing work, however, we seek to leverage a close HR language (Turkish) to facilitate the transfer. The work of Snæbjarnarson et al. (2023) is conceptually most similar; they, however, target a single LR language (Faroese) from a HR family (Germanic branch of the Indo-European family) with many HR relatives (Scandinavian languages).

The three mentioned lines of work typically propose methods to improve XLT starting from a single, given source language (usually EN). Complementary to these lines of work, the work of Lin et al. (2019) and Glavaš and Vulić (2021) instead focus on identifying the best source languages to transfer from for a given target language. Their work considers linguistic and dataset related factors beyond the sole language family. Their findings are complementary to our work, suggesting that even for LR languages that do not have a closely related HR language within their family, it might still be possible to infer such a closely related HR language from another language family.

7 Conclusion

In this work, we contribute to the body of evaluation resources for low-resource (LR) cross-lingual transfer (XLT) by introducing Kardeş-NLU, an evaluation benchmark covering three NLU tasks (NLI, STS, and COPA)—for five Turkic languages: Azerbaijani, Kazakh, Kyrgyz, Uyghur, and Uzbek. Kardeş-NLU allows investigation of an understudied XLT approach: leveraging a high-resource (HR) language to improve transfer to linguistically and genealogically related LR languages. We extend existing intermediate training and fine-tuning approaches for improving LR XLT to integrate Turkish as the HR "sibling" of the Kardeş-NLU languages. Through comprehensive experimentation and analysis, we demonstrated that adding Turkish in task-specific fine-tuning can provide significant XLT gains for Kardeş-NLU languages that are further amplified by incorporating Turkish in bilingual intermediate training strategies. What is more, we also find that Kardeş-NLU is a difficult benchmark for few-shot XLT, observing that established few-shot transfer methods are not effective. Finally, we evaluated two cutting-edge instruction-tuned large language models—mT0 and chatGPT—on Kardeş-NLU, showing that their (zero-shot) performance is inferior on lower-resourced Kardeş-NLU languages (Uyghur, Uzbek, Kyrgyz) and greatly varies across tasks. This proves that there is still a long way to (truly) multilingual NLP. In our subsequent efforts, we will not only seek to extend Kardeş-NLU with additional LR Turkic languages, but also explore how to leverage HR siblings in LR XLT for other language families.

8 Limitations

We strove for both a representative NLU benchmark for Turkic languages and a comprehensive study of XLT to LR target languages with the help of a closely related HR language. Nonetheless, our work is limited in several aspects. Out of 23 live Turkic languages, Kardeş-NLU covers only five. Two main factors determined the set of initially included languages: a limited annotation budget and the ability to find native speakers. The latter is why we ended up with languages that are among the largest Turkic languages in terms of number of native speakers (Kyrgyz, as the smallest, has ca. 5M native speakers). Further, there is a mismatch between the more common Arabic script used for Uyghur and the Cyrillic script we use for it in Kardeş-NLU because our Uyghur annotator was unfamiliar with the Arabic script.

The Kardeş-NLU benchmark is obtained through automatic translations from the existing English test sets to the target languages. This is followed by manual annotation and curation through native speakers to ensure high quality. In order to have suitable translations for culture specific concepts, we instructed our annotators to pay special attention to the idiomaticity of the English sentences during the editing. Despite our best efforts, the resulting datasets might not perfectly reflect the cultural and social elements of the target low-resource languages since their content is tied to original English datasets.

Next, we employed Wikipedias as corpora for our intermediate pretraining. Albeit curated, Wikipedia content is subject to biased, missing or simply incorrect information that can lead to undesired behavior in the resulting models.

Concerning the methodology, we limited our study exclusively to mainstream approaches: (i) intermediate LM-ing for improving the representational quality of mmLMs for a language of interest and (ii) established protocols for downstream zero-shot and few-shot XLT. We acknowledge the existence of more sophisticated (and more recent) XLT methods based, e.g., on gradient manipulation (Wang and Tsvetkov, 2021; Xu and Murray, 2022) or dedicated representational alignment of lexical units (i.e., embedding spaces) (Minixhofer et al., 2022). We hope the research community will use Kardeş-NLU to evaluate and profile existing and future state-of-the-art XLT approaches.

Finally, for the prompt-based evaluation of LLMs, we experiment only with a single instruction (i.e., prompt) adapted from Ahuja et al. (2023). It is reasonable to expect that some prompt engineering effort yields better results.

Acknowledgements

This work was partially supported by DFG (grant SCHU 2246/14-1).

References

David Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba Alabi, Shamsuddeen Muhammad, Peter Nabende, Cheikh M. Bamba Dione, Andiswa Bukula, Rooweither Mabuya, Bonaventure F. P. Dossou, Blessing Sibanda, Happy Buzaaba, Jonathan Mukiibi, Godson Kalipe, Derguene Mbaye, Amelia Taylor, Fatoumata Kabore, Chris Chinenye Emezue, Anuoluwapo Aremu, Perez Ogayo, Catherine Gitau, Edwin Munkoh-Buabeng, Victoire Memdjokam Koagne, Allahsera Auguste Tapo, Tebogo Macucwa, Vukosi Marivate, Mboning Tchiaze Elvis, Tajuddeen Gwadabe, Tosin Adewumi, Orevaoghene Ahia, Joyce Nakatumba-Nabende, Neo Lerato Mokono, Ignatius Ezeani, Chiamaka Chukwuneke, Mofetoluwa Oluwaseun Adeyemi, Gilles Quentin Hacheme, Idris Abdulmumin, Odunayo Ogundepo, Oreen Yousuf, Tatiana Moteu, and Dietrich Klakow. 2022. MasakhaNER 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4488–4508, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

Divyanshu Aggarwal, Vivek Gupta, and Anoop Kunchukuttan. 2022. IndicXNLI: Evaluating multilingual inference for Indian languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10994–11006, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. Mega: Multilingual evaluation of generative ai.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ruth-Ann Armstrong, John Hewitt, and Christopher Manning. 2022. JamPatoisNLI: A jamaican patois natural language inference dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5307–5320, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Akari Asai, Sneha Kudugunta, Xinyan Velocity Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. 2023. Buffet: Benchmarking large language models for few-shot cross-lingual transfer. *arXiv preprint arXiv:2305.14857*.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034.

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.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Goran Glavaš and Ivan Vulić. 2021. Climbing the tower of treebanks: Improving low-resource dependency parsing via hierarchical source selection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4878–4888.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2021. Explicit alignment objectives for multilingual bidirectional encoders. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3633–3643, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multitask benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Chia-Chien Hung, Anne Lauscher, Simone Ponzetto, and Goran Glavaš. 2022. DS-TOD: Efficient domain specialization for task-oriented dialog. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 891–904, Dublin, Ireland. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Kaliyaperumal Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: An empirical study. In *International Conference on Learning Representations*.

Simran Khanuja, Sebastian Ruder, and Partha Talukdar. 2023. Evaluating the diversity, equity, and inclusion of NLP technology: A case study for Indian languages. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1763–1777, Dubrovnik, Croatia. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Jamshidbek Mirzakhalov, Anoop Babu, Duygu Ataman, Sherzod Kariev, Francis Tyers, Otabek Abduraufov, Mammad Hajili, Sardana Ivanova, Abror Khaytbaev, Antonio Laverghetta Jr., Bekhzodbek Moydinboyev, Esra Onal, Shaxnoza Pulatova, Ahsan Wahab, Orhan Firat, and Sriram Chellappan. 2021. A large-scale study of machine translation in Turkic languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5876–5890, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani, Sebastian Ruder, Ibrahim Sa'id Ahmad, Idris Abdulmumin, Bello Shehu Bello, Monojit Choudhury, Chris Chinenye Emezue, Saheed Salahudeen Abdullahi, Anuoluwapo Aremu, Alípio Jorge, and Pavel Brazdil. 2022. NaijaSenti: A Nigerian Twitter sentiment corpus for multilingual sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 590–602, Marseille, France. European Language Resources Association.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1791–1799, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2022. Don't stop fine-tuning: On training regimes for few-shot cross-lingual transfer with multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10725–10742, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2023a. Free lunch: Robust cross-lingual transfer via model checkpoint averaging. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5712–5730, Toronto, Canada. Association for Computational Linguistics.

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2023b. One for all & all for one: Bypassing hyperparameter tuning with model averaging for cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12186–12193.

Vésteinn Snæbjarnarson, Annika Simonsen, Goran Glavaš, and Ivan Vulić. 2023. Transfer to a low-resource language via close relatives: The case study on Faroese. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 728–737, Tórshavn, Faroe Islands. University of Tartu Library.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Zirui Wang and Yulia Tsvetkov. 2021. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Pascale Fung, Timothy Baldwin, Jey Han Lau, Rico Sennrich, and Sebastian Ruder. 2023. NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 815–834, Dubrovnik, Croatia. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Haoran Xu and Kenton Murray. 2022. Por qué não utiliser alla språk? mixed training with gradient optimization in few-shot cross-lingual transfer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2043–2059, Seattle, United States. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. A closer look at few-shot crosslingual transfer: The choice of shots matters. 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 5751–5767, Online. Association for Computational Linguistics.

A LLMs: mT0 and ChatGPT

For mT0, we only use the instance-based prompts, without the task instruction, following Ahuja et al. (2023) (and accept exact matches as correct answers only):

NLI. {PREMISE} *Question:* {HYPOTHESIS} *True, False, or Neither?*

COPA. {PREMISE} {% if question == "cause" %} *This happened because...* {% else %} *As a consequence...* {% endif %} *Help me pick the more plausible option:* -{CHOICE1}-{CHOICE2}

For ChatGPT, we slightly modify the prompts from Ahuja et al. (2023) due to the fact that they perform in-context few-shot learning, whereas we carry out zero-shot prediction:

NLI. *You are an NLP assistant whose purpose is to solve Natural Language Inference (NLI) problems. NLI is the task of determining the inference relation between two (short, ordered) texts. For the given two sentences, you need to predict one of the following: 1. Entailment, 2. Contradiction, or 3. Neither (Neutral). Sentence 1:* {PREMISE}. *Sentence 2:* {HYPOTHESIS}. *Answer:*

COPA. *You are an AI assistant whose purpose is to perform open-domain commonsense causal reasoning. You will be provided a premise and two alternatives, where the task is to select the alternative that more plausibly has a causal relation with the premise. Answer as concisely as possible.* PREMISE {% if question == "cause" %} *This happened because...* {% else %} *As a consequence...* {% endif %}: *Alternative 1:* CHOICE1 *Alternative 2:* CHOICE2

For NLI, the model's output is compared directly against the target label (*True*, *False*, or *Neither*). For COPA, it is compared against the correct alternative ({CHOICE1} or {CHOICE2}). Since the models are free to generate any text, they can theoretically perform below the random baseline (33% for NLI and 50% for COPA).

Table 5 displays per language and average results for zero-shot evaluations on NLI and COPA for the XLM-R base versions that we experiment with, mT0 of various sizes, and ChatGPT. We also experiment with the templates that are translated to the target language using Google Translate. However, those versions overall performed worse than the English versions, most likely because of the low translation quality. We can see that mT0's performance on COPA improves drastically when it is scaled to XL and XXL versions. It should be noted that mT0's instruction tuning dataset includes the Social IQA dataset, which is similar to the COPA dataset. This might explain the larger model's strong performance on this dataset outperforms zero-shot XLM-R variants.

B Computational Resources

All the experiments were run on a single V100 with 32GB VRAM. We roughly estimate that total GPU time accumulates to 2800 hours across all experiments.

C Adapter Fine-Tuning Experiments

In preliminary experiments, we investigated the adapter-based equivalents to TLLM and BALM (on STS and NLI) (Pfeiffer et al., 2020; Parović et al., 2022). We report per-language and averaged scores in Table 6. Full fine-tuning of the mmLM outperformed the adapter-based tuning, especially on lower-resourced languages.

Target Language LM-ing Adapters (TLLM-AD). We first train monolingual language adapters on target languages via MLM-ing. We then stack a task adapter on top and fine-tune it on the corresponding downstream data—English, Turkish or English and Turkish jointly—while keeping the language adapter frozen.

Bilingual Alternating LM-ing Adapters (BALM-AD). Here, we stick to Parović et al. 2022 and update the language adapter´s parameters alternately by one batch on the target language data followed by one batch on Turkish data. Afterwards, we fine-tune task adapters on either English, Turkish or English and Turkish jointly, while keeping the language adapter frozen.

Adapter Training Details. We trained monolingual language adapters for 25000 steps and bilingual ones for 50000. We set the learning rate to $1e{-}4$ and the batch size to 64. For task adapters, we applied the same hyperparameters used for our full fine-tuning experiments explained in section 4 but lowered the learning rate to $1e{-}4$, as suggested by Pfeiffer et al. 2020.

		Azerbaijani			Kazakh			Kyrgyz			Uyghur			Uzbek			Average		
		EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR
NLI	Base	76.5	**80.1**	79.6	73.8	76.3	**77.3**	70.4	73.9	74.1	42.2	44.4	42.9	70.7	72.0	71.8	66.7	69.4	69.1
	TLM	77.3	79.0	79.2	75.3	76.3	76.8	72.4	74.1	74.4	56.7	57.1	56.9	73.1	74.3	74.8	71.0	72.2	72.4
	BALM	77.3	78.8	79.3	74.4	75.3	77.0	71.6	73.4	74.0	57.4	58.7	58.0	73.1	74.5	75.0	70.8	72.1	72.7
	BJLM	76.4	78.4	79.3	74.9	75.1	76.8	71.9	74.3	**75.5**	57.2	59.2	**59.4**	73.4	74.6	**75.7**	70.7	72.3	**73.3**
	mT0$_{small}$	35.3			34.9			36.8			36.6			35.3			35.8		
	mT0$_{base}$	40.5			40.3			39.8			38.3			40.4			39.8		
	mT0$_{large}$	40.8			42.5			42.0			41.9			41.2			41.7		
	mT0$_{XL}$	56.9			55.7			53.0			49.4			55.6			54.1		
	mT0$_{XXL}$	60.7			59.4			58.1			54.3			58.9			58.2		
	chatGPT	56.4			48.0			47.1			47.7			47.9			49.4		
COPA	Base	60.1	61.1	60.9	60.7	60.8	59.9	59.7	60.0	59.4	51.8	52.7	52.7	57.3	59.5	60.1	57.9	58.8	58.6
	TLM	62.1	62.1	61.5	55.7	55.8	56.1	57.5	59.7	58.9	49.9	50.3	49.3	62.9	63.2	62.5	57.6	58.2	57.7
	BALM	57.2	58.3	59.4	59.1	59.5	59.7	56.1	59.9	59.1	51.1	53.9	52.5	60.5	61.7	61.9	56.8	58.6	57.9
	BJLM	61.8	63.3	63.3	58.4	58.6	57.7	56.8	61.5	**62.0**	50.9	52.2	53.9	61.7	60.5	62.9	57.9	59.2	60.0
	mT0$_{small}$	34.2			7.6			3.4			5.6			43.6			18.8		
	mT0$_{base}$	32.0			3.6			5.8			4.2			39.8			17.1		
	mT0$_{large}$	38.0			38.2			30.4			24.2			38.4			33.8		
	mT0$_{XL}$	60.4			62.8			50.4			47.6			63.2			56.9		
	mT0$_{XXL}$	**81.2**			**74.6**			57.8			**61.4**			**80.6**			**71.1**		
	chatGPT	73.0			63.4			56.6			57.0			55.6			61.1		

Table 5: Zero-Shot results for the target languages and the average results across the five languages for XLM-R base, mT0 and chatGPT models. The best results for each language-task pair are shown in **bold**.

		Azerbaijani			Kazakh			Kyrgyz			Uyghur			Uzbek			Average		
		EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR	EN	TR	EN,TR
NLI	TLLM	77.3	79.0	79.2	75.3	76.3	76.8	72.4	74.1	74.4	56.7	57.1	56.9	73.1	74.3	74.8	71.0	72.2	72.4
	BALM	77.3	78.8	79.3	74.4	75.3	77.0	71.6	73.4	74.0	57.4	58.7	**58.0**	73.1	74.5	**75.0**	70.8	72.1	**72.7**
	TLLM-AD	77.1	78.2	**80.3**	74.0	74.8	76.8	70.1	72.7	74.5	48.3	47.0	48.3	71.1	71.1	73.4	68.1	68.8	70.6
	BALM-AD	77.9	78.0	80.1	73.3	75.2	**77.6**	70.7	73.2	**74.7**	47.8	46.4	46.8	70.5	71.8	73.1	68.1	69.0	70.5
STS	TLLM	75.8	75.5	78.1	80.6	80.1	81.9	71.3	71.8	74.2	70.6	69.3	71.3	70.6	67.0	76.9	73.8	72.7	76.5
	BALM	72.7	78.7	79.7	81.4	83.2	83.9	71.1	77.3	**78.3**	72.8	72.3	**73.5**	72.5	77.6	**79.3**	74.1	77.8	**79.0**
	TLLM-AD	76.1	77.5	79.5	82.0	81.4	**84.3**	74.0	75.4	77.8	69.7	68.4	70.5	75.2	75.5	77.4	75.4	75.6	77.9
	BALM-AD	76.2	77.5	**79.9**	82.3	81.6	84.1	73.2	75.5	77.3	68.2	67.3	70.0	75.1	75.0	77.3	75.1	75.4	77.7

Table 6: Zero-Shot XLT results on Kardeş-NLU (NLI and STS) for two adapter strategies (TLLM-AD and BALM-AD) and source fine-tuning datasets (English only, Turkish only, and English and Turkish combined). The best results for each language-task pair are shown in **bold**.

D Few-Shot Results

		Zero-Shot			Few-Shot																	
					Sequential									Joint								
		EN	TR	EN,TR	EN			TR			EN,TR			EN			TR			EN,TR		
Shots		-	-	-	10	50	100	10	50	100	10	50	100	10	50	100	10	50	100	10	50	100
Azerbaijani	Base	76.5	80.1	79.6	73.3	76.6	76.3	74.9	78.5	77.9	75.2	78.8	79.0	75.0	74.7	74.1	77.7	76.9	76.8	76.7	77.1	77.3
	TLM	77.3	79.0	79.2	75.7	77.7	77.8	75.7	78.7	79.3	76.9	79.1	78.9	76.4	77.0	76.7	77.8	77.7	77.2	78.0	78.3	78.2
	BALM	77.3	79.0	79.2	75.4	77.2	77.3	76.5	78.1	78.1	76.7	78.9	79.2	74.8	76.0	76.3	78.0	78.4	78.1	77.6	77.5	78.0
	BJLM	77.3	78.8	79.3	72.3	77.5	77.3	75.8	78.7	78.3	77.3	79.1	79.2	76.6	76.9	75.7	77.8	78.2	77.3	78.3	78.4	77.7
Kazakh	Base	73.8	76.3	77.3	69.7	73.6	73.5	72.0	75.0	75.3	73.3	75.5	76.0	71.1	71.5	71.4	74.3	73.0	72.7	74.6	74.4	74.3
	TLM	75.3	76.3	76.8	72.4	75.5	76.3	75.1	75.9	75.7	74.8	76.8	76.1	73.8	75.2	74.8	75.2	75.6	74.6	76.0	75.8	76.4
	BALM	74.4	75.3	77.0	72.8	75.3	74.7	72.9	75.8	75.7	75.1	76.4	76.9	73.8	73.8	74.5	74.6	74.8	74.2	74.9	74.7	75.8
	BJLM	74.9	75.1	76.8	73.2	74.8	75.0	73.0	74.5	74.6	74.5	76.8	76.4	73.3	74.1	73.6	74.1	75.0	74.3	75.2	75.2	74.7
Kyrgyz	Base	70.4	73.9	74.1	66.6	70.6	70.5	69.4	72.3	72.7	70.3	73.1	73.6	68.9	69.7	69.2	70.7	69.4	69.5	70.8	70.5	71.7
	TLM	72.4	74.1	74.4	71.0	73.6	73.1	72.2	73.6	74.0	72.9	75.4	75.4	71.4	71.6	71.9	72.4	73.4	72.6	72.8	73.0	73.2
	BALM	71.6	73.4	74.0	69.2	73.2	72.6	71.2	73.4	73.0	73.0	74.5	74.7	71.0	71.4	71.8	71.7	72.3	71.9	73.0	73.2	73.0
	BJLM	71.9	74.3	75.5	71.7	73.1	73.3	72.9	74.0	73.5	73.7	75.8	75.7	72.0	72.8	72.0	73.4	72.8	73.6	72.6	73.6	73.8
Uyghur	Base	42.2	44.4	42.9	41.5	49.2	50.1	45.0	47.9	50.5	43.5	48.6	49.6	43.2	47.8	49.9	43.8	48.4	49.8	42.2	47.9	48.3
	TLM	56.7	57.1	56.9	50.1	53.7	58.0	52.1	57.3	58.8	55.3	56.8	57.9	52.6	54.6	56.6	52.9	56.5	56.2	52.4	55.7	58.1
	BALM	57.4	58.7	58.0	51.4	57.0	58.3	53.0	58.0	59.5	51.9	58.3	59.4	53.7	56.3	55.8	54.9	57.9	58.9	54.0	56.4	57.4
	BJLM	57.2	59.2	59.4	51.1	56.4	57.8	52.8	57.3	57.3	51.6	57.0	58.8	52.8	54.4	55.9	54.5	56.4	57.1	54.0	56.1	57.9
Uzbek	Base	70.7	72.0	71.8	66.5	69.5	69.8	67.1	71.6	70.2	67.6	71.3	72.3	66.5	67.5	67.4	68.6	69.0	68.6	67.9	68.6	69.0
	TLM	73.1	74.3	74.8	71.3	73.3	73.4	71.3	74.1	73.9	73.1	74.9	74.4	72.4	73.1	73.3	72.4	73.2	72.9	72.7	73.2	73.5
	BALM	73.1	74.5	75.0	70.9	71.6	73.4	71.4	73.9	73.8	73.3	74.7	75.1	72.1	72.4	73.5	73.4	73.9	73.2	73.1	73.2	73.7
	BJLM	73.4	74.6	75.7	69.3	73.1	73.3	71.4	74.0	74.0	72.2	74.8	75.0	72.4	73.4	72.3	73.4	74.1	73.7	73.1	74.0	75.1

Table 7: Per-language results of *sequential* and *joint* transfer on Kardeş-NLI.

		Zero-Shot			Few-Shot																	
					Sequential									Joint								
		EN	TR	EN,TR	EN			TR			EN,TR			EN			TR			EN,TR		
Shots		-	-	-	10	50	100	10	50	100	10	50	100	10	50	100	10	50	100	10	50	100
Azerbaijani	Base	60.1	61.1	60.9	62.3	62.5	63.8	61.5	61.3	62.5	61.9	62.3	62.5	60.3	62.2	61.9	62.3	62.8	62.7	61.7	62.8	62.9
	TLM	62.1	62.1	61.5	60.1	60.7	60.6	60.3	60.3	62.1	59.9	60.8	61.1	60.8	61.2	62.1	62.3	60.8	60.6	61.6	61.7	62.6
	BALM	57.2	58.3	59.4	58.5	58.3	59.2	58.8	58.0	59.2	60.1	58.7	59.8	59.5	59.8	57.7	58.9	59.3	59.1	62.7	60.6	59.3
	BJLM	61.8	63.3	63.3	61.1	62.4	62.1	62.5	61.9	62.9	61.0	62.1	61.7	62.0	62.8	61.9	62.1	63.7	61.9	61.9	62.3	62.4
Kazakh	Base	60.7	60.8	59.9	55.6	59.3	60.1	57.6	60.7	60.3	56.7	60.4	60.3	58.7	59.2	60.8	60.2	60.7	60.9	60.7	60.8	61.9
	TLM	55.7	55.8	56.1	54.4	56.1	57.2	54.8	55.5	57.9	54.9	56.5	57.9	55.4	56.4	56.5	56.3	57.6	58.4	56.6	58.3	59.5
	BALM	59.1	59.5	59.7	58.6	59.4	60.3	55.9	59.5	59.5	57.1	58.7	59.9	57.5	57.9	60.3	60.0	59.3	59.8	59.9	60.7	59.3
	BJLM	58.4	58.6	57.7	56.0	57.9	60.1	58.3	58.9	60.5	58.3	59.5	60.5	57.5	59.8	58.9	58.5	59.5	59.2	59.6	59.8	59.7
Kyrgyz	Base	59.7	60.0	59.4	56.6	59.0	59.7	58.0	58.5	59.0	59.3	59.3	59.7	60.1	60.1	61.1	61.1	60.5	60.2	61.3	61.1	61.1
	TLM	57.5	59.7	58.9	58.5	58.9	61.2	59.7	60.9	61.9	58.7	60.0	60.2	58.7	58.2	59.7	60.1	60.6	59.5	61.3	61.5	61.7
	BALM	56.1	59.9	59.1	57.6	58.1	58.3	58.1	61.7	60.7	57.6	59.8	60.3	56.1	58.1	57.7	60.7	61.7	60.1	58.5	60.9	58.9
	BJLM	56.8	61.5	62.0	57.3	59.5	60.8	60.5	63.1	61.3	60.1	62.4	62.1	59.5	59.3	60.1	61.3	61.9	62.3	62.2	62.9	60.9
Uyghur	Base	51.8	52.7	52.7	51.7	50.7	52.5	51.3	50.3	51.9	50.7	51.3	51.7	51.3	50.9	52.4	51.1	50.5	50.1	51.5	50.6	51.7
	TLM	49.9	50.3	49.3	50.9	48.1	50.5	48.6	49.1	52.7	48.7	49.7	51.1	49.2	49.9	50.2	49.9	49.9	50.4	49.5	49.8	52.3
	BALM	51.1	53.9	52.5	51.1	49.4	50.7	53.3	51.2	51.7	52.9	51.2	50.7	50.8	50.9	49.6	54.2	52.5	51.5	52.5	52.5	51.7
	BJLM	50.9	52.2	53.9	50.7	49.9	51.5	49.7	50.6	51.6	49.5	50.7	52.4	50.6	50.1	50.5	51.0	51.9	51.4	52.9	51.9	51.7
Uzbek	Base	57.3	59.5	60.1	55.9	57.9	57.6	55.7	57.1	57.1	56.6	55.9	57.1	57.3	57.2	58.7	58.9	58.0	58.6	59.5	59.6	59.7
	TLM	62.9	63.2	62.5	59.9	63.1	62.7	62.1	63.5	63.1	61.1	62.8	64.1	62.1	61.7	63.1	61.9	64.7	64.1	63.9	63.7	62.8
	BALM	60.5	61.7	61.9	56.9	60.7	62.3	58.2	59.8	61.3	60.3	61.4	61.2	60.3	62.3	60.6	61.3	60.9	60.3	61.7	62.3	62.1
	BJLM	61.7	60.5	62.9	60.7	63.3	62.1	59.3	61.9	62.4	61.2	64.2	62.3	60.9	61.9	62.7	61.5	62.3	61.7	63.9	62.7	64.4

Table 8: Per-language results of *sequential* and *joint* few-shot transfer on Kardeş-COPA.

		Zero-Shot			Few-Shot																	
					Squential									Joint								
		EN	TR	EN,TR	EN			TR			EN,TR			EN			TR			EN,TR		
Shots		-	-	-	10	50	100	10	50	100	10	50	100	10	50	100	10	50	100	10	50	100
Azerbaijani	Base	80.3	78.9	80.4	74.5	76.7	76.9	75.7	77.2	77.0	77.6	78.8	78.2	79.3	78.8	79.2	79.7	80.2	80.0	80.4	80.8	80.8
	TLM	75.8	75.5	78.1	75.0	76.2	76.3	75.1	76.6	77.2	77.5	78.0	78.9	77.5	77.4	78.0	76.2	77.4	77.9	78.8	79.2	79.7
	BALM	72.7	78.7	79.7	75.6	76.3	76.3	76.0	77.2	78.1	77.6	78.7	79.4	75.8	76.4	77.1	79.4	79.6	80.1	80.1	80.6	80.5
	BJLM	69.3	77.0	78.3	73.9	74.8	75.6	76.6	77.5	77.9	77.3	78.2	78.5	75.3	75.9	76.4	78.1	79.1	79.5	79.6	80.2	80.5
Kazakh	Base	85.8	84.1	84.8	81.6	82.1	82.4	81.2	82.3	82.3	82.5	83.1	83.8	84.5	84.4	84.9	84.5	85.1	85.4	85.0	85.6	85.6
	TLM	80.6	80.1	81.9	81.1	82.0	82.2	81.2	81.2	81.9	82.5	84.0	83.8	81.8	83.2	83.5	80.9	82.6	83.3	82.6	84.0	84.3
	BALM	81.4	83.2	83.9	81.5	82.7	82.6	82.0	83.2	84.3	82.5	84.6	84.4	82.6	83.7	84.2	83.9	84.7	85.0	84.7	85.6	85.9
	BJLM	78.6	83.2	84.6	79.6	81.5	82.0	80.9	83.1	83.3	82.4	83.7	84.5	80.5	82.3	82.6	83.9	84.5	84.9	85.1	85.6	85.8
Kyrgyz	Base	78.2	77.9	78.7	71.3	72.1	73.3	73.7	74.7	73.4	74.0	75.1	75.9	76.4	76.0	75.8	78.7	79.5	79.4	78.8	79.8	79.5
	TLM	71.3	71.8	74.2	71.2	70.8	71.6	72.5	73.6	73.4	73.4	73.2	73.6	72.7	73.8	73.8	74.1	75.7	76.8	76.0	77.2	77.1
	BALM	71.1	77.3	78.3	69.4	71.3	72.3	74.5	76.5	75.5	75.7	77.0	75.4	72.3	72.8	73.6	77.7	78.6	78.4	78.1	78.7	79.3
	BJLM	69.9	75.1	77.3	68.8	70.6	72.4	73.6	75.0	74.1	74.8	75.8	76.1	71.7	73.3	74.3	76.4	77.2	76.9	77.4	77.9	78.0
Uyghur	Base	69.2	64.8	64.2	65.7	71.2	69.2	67.4	71.8	69.7	66.1	71.1	70.9	64.7	71.1	71.3	64.2	70.9	70.9	63.7	70.0	71.5
	TLM	70.6	69.3	71.3	68.4	71.8	72.4	71.5	72.6	72.0	71.9	73.0	73.8	69.3	72.5	72.6	69.6	72.1	72.7	70.8	73.2	73.6
	BALM	72.8	72.3	73.5	71.5	74.1	74.3	72.8	74.2	74.2	73.2	74.5	74.8	71.3	74.7	74.6	71.7	74.9	75.0	72.9	75.3	75.6
	BJLM	65.7	66.9	69.0	69.0	72.7	71.7	70.5	72.1	71.4	70.4	73.2	73.1	68.5	73.3	73.2	68.3	72.4	72.4	69.8	73.7	73.7
Uzbek	Base	78.3	77.2	77.1	74.2	75.4	75.2	74.6	76.2	75.7	76.6	77.6	76.7	76.7	77.5	77.1	77.9	78.7	78.5	77.8	78.8	78.9
	TLM	70.6	67.0	76.9	72.5	75.6	75.5	74.2	75.6	76.1	77.0	78.2	78.0	74.1	77.0	76.7	75.4	77.2	77.2	77.8	79.0	79.2
	BALM	72.5	77.6	79.3	74.4	75.7	76.1	75.9	76.9	76.9	77.4	78.1	78.1	75.4	77.2	77.6	78.6	79.3	79.3	79.9	80.3	80.5
	BJLM	71.1	76.8	77.3	72.6	74.7	75.2	74.5	76.8	77.3	75.7	77.8	78.1	74.0	76.1	76.4	77.1	78.5	78.7	77.8	79.0	79.1

Table 9: Per-language results of *sequential* and *joint* few-shot transfer on Kardeş-STS.

Chapter 3

Graph Algorithms for Multiparallel Word Alignment

Graph Algorithms for Multiparallel Word Alignment

Ayyoob Imani[*1], **Masoud Jalili Sabet**[*1], **Lütfi Kerem Şenel**[1],
Philipp Dufter[1], **François Yvon**[2], **Hinrich Schütze**[1]
[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2]Université Paris-Saclay, CNRS, LISN, France
{ayyoob, masoud, lksenel, philipp}@cis.lmu.de,
francois.yvon@limsi.fr

Abstract

With the advent of end-to-end deep learning approaches in machine translation, interest in word alignments initially decreased; however, they have again become a focus of research more recently. Alignments are useful for typological research, transferring formatting like markup to translated texts and can be used in the decoding of machine translation systems. At the same time, massively multilingual processing is becoming an important NLP scenario and pretrained language and machine translation models that are truly multilingual are proposed. However, most alignment algorithms rely on bitexts only and do not leverage the fact that many parallel corpora are multiparallel. In this work, we exploit multiparallelity of corpora by representing an initial set of bilingual alignments as a graph and then predicting additional edges in the graph. We present two graph algorithms for edge prediction: one inspired by recommender systems and one based on network link prediction. Our experimental results show absolute improvements of F_1 of up to 28% over the baseline bilingual word aligner in different datasets.

1 Introduction

Word alignment is a challenging NLP task that plays an essential role in statistical machine translation and is useful for neural machine translation (Alkhouli and Ney, 2017; Alkhouli et al., 2016; Koehn et al., 2003). Other applications of word alignments include bilingual lexicon induction, annotation projection, and typological analysis (Shi et al., 2021; Rasooli et al., 2018; Müller, 2017; Lewis and Xia, 2008). With the advent of deep learning, interest in word alignment initially decreased. However, recently a new wave of publications has again drawn attention to the task (Jalili Sabet et al., 2020; Dou and Neubig, 2021; Marchisio et al., 2021; Wu and Dredze, 2020).

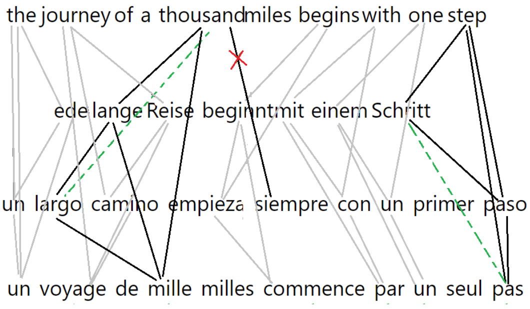

the journey of a thousand miles begins with one step
ede lange Reise beginnt mit einem Schritt
un largo camino empieza siempre con un primer paso
un voyage de mille milles commence par un seul pas

Figure 1: Bilingual alignments of a verse in English, German, Spanish, and French. Two of the alignment edges not found by the bilingual method are German "Schritt" to French "pas" and Spanish "largo" to English "thousand miles". By looking at the structure of the entire graph, one can infer the correctness of these two edges.

In this paper we propose *MPWA* (MultiParallel Word Alignment), a framework that employs graph algorithms to exploit the information latent in a multiparallel corpus to achieve better word alignments than aligning pairs of languages in isolation. Starting from translations of a sentence in multiple languages in a multiparallel corpus, MPWA generates bilingual word alignments for all language pairs using any available bilingual word aligner. MPWA then improves the quality of word alignments for a target language pair by inspecting how they are aligned to other languages. *The central idea is to exploit the graph structure of an initial multiparallel word alignment to improve the alignment for a target language pair.* To this end, MPWA casts the multiparallel word alignment task as a link (or edge) prediction problem. We explore standard algorithms for this purpose: Adamic-Adar and matrix factorization. While these two graph-based algorithms are quite different and are used in different applications, we will show that MPWA effectively leverages them for high-performing word alignment.

[*] Equal contribution - random order.

Link prediction methods are used to predict whether there should be a link between two nodes in a graph. They have various applications like movie recommendations, knowledge graph completion, and metabolic network reconstruction (Zhang and Chen, 2018). We use the Adamic-Adar index (Adamic and Adar, 2003); it is a second-order link prediction algorithm, i.e., it exploits the information of neighbors that are up to two hops aways from the starting target nodes (Zhou et al., 2009). We use a second-order algorithm since a set of aligned words in multiple languages (representing a concept) tends to establish a clique (Dufter et al., 2018). This means that exploring the influence of nodes at a distance of two in the graph provides informative signals while at the same time keeping runtime complexity low.

Matrix factorization is a collaborative filtering algorithm that is most prominently used in recommender systems where it provides users with product recommendations based on their interactions with other products. This method is especially useful if the matrix is sparse (Koren et al., 2009). This is true for our application: Given two translations of a sentence with lengths M and N, among all possible alignment links ($M \times N$), only a few ($O(M + N)$) are correct. This is partly due to fertility: words in the source language generally have only a few possible matches in the target language (Zhao and Gildea, 2010).

A multiparallel corpus provides parallel sentences in more than two languages. This type of corpus facilitates the study of multiple languages together, which is especially important for research on low resource languages. As far as we know, out of all available multiparallel corpora, the Parallel Bible Corpus (Mayer and Cysouw, 2014) (PBC) provides the highest language coverage, supporting 1334 different languages, many of which belong to categories 0 and 1 (Joshi et al., 2020) – that is, they are languages for which no language technologies are available and that are severely underresourced.

MPWA has especially strong word alignment improvements for distant language pairs for which existing bilingual word aligners perform poorly. Much work that addresses low resource languages relies on the availabiliy of monolingual corpora. Complementarily, MPWA assumes the existence of a very small (a few 10,000s of sentences in our case) parallel corpus and then takes advantage of information from the other languages in the parallel corpus. This is an alternative approach that is especially important for low resource languages for which monolingual data often are not available.

The PBC corpus does not contain a word alignment gold standard. To conduct the comparative evaluation of our new method, we port three existing word alignment gold standards of Bible translations to PBC, for the language pairs English-French, Finnish-Hebrew and Finnish-Greek. We also create artificial multiparallel datasets for four widely used word alignment datasets using machine translation. We evaluate our method with all seven datasets. Results demonstrate substantial improvements in all scenarios.

Our main contributions are:

1. We propose two graph-based algorithms for link prediction (i.e., the prediction of word alignment edges in the alignment graph), one based on second-order link prediction and one based on recommender systems for improving word alignment in a multiparallel corpus and show that they perform better than established baselines.

2. We port and publish three word alignment gold standards for the Parallel Bible Corpus.

3. We show that our method is also applicable, using machine translation, to scenarios where multiparallel data is not available.

4. We publish our code[1] and data.

2 Related Work

Bilingual Word Aligners take different approaches. Some are based on statistical analysis, like IBM models (Brown et al., 1993), Giza++ (Och and Ney, 2003a), fast-align (Dyer et al., 2013) and Eflomal (Östling and Tiedemann, 2016). Another more recent group, including SimAlign (Jalili Sabet et al., 2020) and Awesome-align (Dou and Neubig, 2021), utilizes neural language models. The last group is based on neural machine translation (Garg et al., 2019; Zenkel et al., 2020). While neural models outperform statistical models, for cases where only a small parallel dataset is available, statistical models are still superior. In this paper we use PBC, a corpus with 1334 languages, of which only about two hundred are supported by multilingual language models like Bert and XLM-R (Devlin et al., 2019; Conneau et al., 2020). MPWA can

[1] https://github.com/cisnlp/graph-align

leverage multiparallelism on top of any bilingual word aligner; in this paper, we use Eflomal and SimAlign.

Multiparallel corpus alignment. Most work on word alignment has focused on bilingual corpora. To the best of our knowledge, only one method specifically designed for multiparallel corpora was previously proposed: (Östling, 2014).[2] However this method is outperformed by a "biparallel" method by the same author, Eflomal (Östling and Tiedemann, 2016). We compare with Eflomal in our experiments.

Cohn and Lapata (2007) make use of multiparallel corpora to obtain more reliable translations from small datasets. Kumar et al. (2007) show that multiparallel corpora can be of benefit to reach better performance in phrase-based statistical machine translation (SMT). Filali and Bilmes (2005) present a multilingual SMT-based word alignment model, extending IBM models, based on HMM models and a two step alignment procedure. Since the goal of this research is to tackle word alignment directly without considering machine translation, these works are not considered here.

In another line of research, Lardilleux and Lepage (2008a) introduce a corpus splitting method to come up with a perfect alignment of multiwords. Lardilleux and Lepage (2008b), and Lardilleux and Lepage (2009) suggest to rely only on low frequency terms for a similar purpose: sub-sentential alignment. These methods solve a somewhat different problem than what is addressed by us. Other usages of multiparallel corpora are language comparison (Mayer and Cysouw, 2012), typology studies (Östling, 2015; Asgari and Schütze, 2017; Imani-Googhari et al., 2021) and SMT (Nakov and Ng, 2012; Bertoldi et al., 2008; Dyer et al., 2013)

Matrix factorization and link prediction. Matrix factorization is a technique that factors, in the most typical case, a matrix into two lower-ranked matrices in which the latent factors of the original matrix are represented. Matrix factorization approaches have been widely used in document clustering (Xu et al., 2003; Shahnaz et al., 2006), topic modeling (Kuang et al., 2015; Choo et al., 2013) information retrieval (Zamani et al., 2016; Deerwester et al., 1990) and NLP tasks like word sense disambiguation (Schütze, 1998). In 2009, Netflix's recommender system competition revealed that this technique effectively works for collaborative filtering (Koren et al., 2009). Since then it has been a state of the art method in recommender systems.

Link prediction algorithms are widely used in different areas of science since many social, biological, and information systems can be described as networks with nodes and connecting links (Zhou et al., 2009). Link prediction algorithms compute the likelihood of links based on different heuristics. One can categorize available methods based on the maximum number of hops they consider in their computations for each node (Zhang and Chen, 2018). First order algorithms, such as common neighbors (CN), only consider one hop neighborhoods, e.g., (Barabási and Albert, 1999). Second order methods consider two hops, e.g., (Zhou et al., 2009). Finally, higher order methods take the whole network into account for making predictions (Brin and Page, 1998; Jeh and Widom, 2002; Rothe and Schütze, 2014). In this paper, we use a two-hop method since it offers a good tradeoff between effectiveness and efficiency.

3 Methods

3.1 The MPWA framework

While a bilingual aligner considers each language pair separately, MPWA utilizes the synergy between all language pairs to improve word alignment performance. In Figure 1, Eflomal alignments of a sentence from PBC in four different languages are depicted. Although Eflomal has failed to find the link between German "Schritt" and French "pas", we can easily find this relation by observing that the four nodes "step", "Schritt", "paso", and "pas" are fully connected, except for the edge from "Schritt" to "pas". In this case, the inference amounts to a completion of a clique. However, most cases are not that simple. In the figure, English "thousand miles" is mistakenly aligned to Spanish "siempre" although its alignments to German "lange" and French "mille" are correct. We would like to infer that "thousand miles" should be aligned to "largo", but in this case creating a fully connected subgraph, i.e., a clique (which would include "siempre"), would add many incorrect edges. Given the complexity and error-proneness of initial bilingual alignments, inferring an alignment between two languages from a multiparallel alignment in general is a complex problem.

Starting from a multiparallel corpus, we first generate bilingual alignments for all language pairs.

[2] `https://github.com/robertostling/eflomal`

MPWA then employs a prediction algorithm to find and add new alignment links. In this paper, we focus on two prediction algorithms: non-negative matrix factorization and Adamic-Adar link prediction.

3.2 Non-negative matrix factorization

Non-negative matrix factorization (NMF) has been used in many different applications. After discovery of its effectiveness for collaborative recommendation (Koren et al., 2009), it was widely accepted as a standard method for recommender systems.

In a standard recommender system with m users and n items, ratings (a number from 1 to 5) from each user for the items they have seen so far are known. The aim is to predict the ratings the user would give to unseen items and, based on these predictions, recommend new items to the user. As described by (Luo et al., 2014), let $W = [w_{u,i}] \in \mathbb{R}^{m\times n}$ be the matrix of ratings. For NMF to work it is essential that the matrix be sparse, thus if a user's rating for an item is unknown, the corresponding cell is zeroed. The matrix W is then decomposed into two low-rank non-negative matrices, $T = [t_{u,k}] \in \mathbb{R}^{m\times r}$ and $V = [v_{k,i}] \in \mathbb{R}^{r\times n}$ such that $TV \approx W$ and $r \ll \min(m, n)$. r is a hyperparameter. By multiplication of these two matrices we end up with a reduced matrix $W' = TV$ in which each zeroed cell $w_{u,i}$ from matrix W is replaced with a value $w'_{u,i}$ that represents a prediction for the rating that user u would give to item i. NMF solves the following optimization program:

$$\underset{T,V}{\text{argmin}} \left(\|W - TV\|^2\right)$$
$$\text{subject to} \quad T, V \geqslant 0$$

This optimization problem can be solved by gradient descent using the following updates:

$$t_{u,k} \leftarrow t_{u,k} + \eta_{u,k}((WV^T)_{u,k} - (TVV^T)_{u,k})$$
$$v_{k,i} \leftarrow v_{k,i} + \eta_{k,i}((T^TW)_{k,i} - (T^TTV)_{k,i})$$

In this equation, η is the learning rate. To guarantee non-negativity, it is defined as:

$$\eta_{u,k} = \frac{t_{u,k}}{(TVV^T)_{u,k}}, \quad \eta_{k,i} = \frac{v_{k,i}}{(T^TTV)_{k,i}}$$

	I	can	see	ich	kann	es	sehen	je	vois
I	5		1	5		1		5	1
can		5	1		5		1		
see	1	1	5		1		5	1	5
ich	5	1		5			1	5	1
kann	1	5		1	5				
es						5	1		
sehen	1		5		1		5	1	
je	5	1		5		1		5	1
vois	1		5	1			5	1	5

Figure 2: An example of how the original matrix is filled for a sentence in three languages. Zero entries are left blank for readability.

Note that the objective function only takes account of non-zero cells. Luo et al. (2014) propose an approach that takes advantage of the sparseness of the matrix for faster computation. In addition, Tikhonov regularization is integrated to improve precision, recall, and convergence rate.

We use the implementaion of NMF provided by the Surprise[3] library, with default hyperparameters ($r = 15$, n_epochs $= 50$).

3.2.1 NMF in MPWA framework

We create a separate matrix W for each sentence in the multiparallel corpus. Tokens in the sentence play the role of both users and items, i.e., we consider each token both as a row and as a column. Figure 2 shows an example of a sentence in a toy English-German-French multiparallel corpus. If two tokens are aligned using the bilingual aligner, we fill the corresponding cell with the highest rating (5). To give a few negative examples to the algorithm, if a token x from language L_1 is aligned to token y in language L_2, we pick another random token z from L_2 and fill the corresponding cell of x to z with the lowest rating (1). We zero out all other cells. Next we apply the matrix factorization algorithm to this matrix and then compute the reduced matrix W' from the factors. Now we grab the predicted alignment scores between source and target languages from W'. To extract new alignment edges we apply the Argmax algorithm (Jalili Sabet et al., 2020). Argmax creates an alignment edge between word w_i from language L_1 and word w_j from language L_2 if among all words from L_2, w_i has the highest alignment score with w_j, and among all words from L_1, w_j has the highest alignment score with w_i.

3.3 Link Prediction

A multiparallel sentence annotated with bilingual word alignments can be considered to be a graph with words from all translations as nodes and the

[3] http://surpriselib.com/

word alignments as edges. Link prediction algorithms such as Common Neighbors (CN) and Adamic-Adar (AdAd) estimate the likelihood of having an edge between two nodes x and y in the graph based on the similarity of their neighborhoods. The CN index weights all common neighbors equally. In contrast, AdAd gives higher weight to neighbors with low degrees because sharing a neighbor that in turn has few neighbors is more significant. Therefore, we use the AdAd index. It is defined as:

$$\mathrm{AdAd}_{x,y} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|} \quad (1)$$

where $\Gamma(x)$ is the neighborhood of x.

If we use a word aligner that produces a score for each alignment edge, we can use Weighted Adamic-Adar (Lü and Zhou, 2010):

$$\mathrm{WAdAd}_{x,y} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(z,y)}{\log(1 + S(z))} \quad (2)$$

where $w(x, z)$ is the similarity score of x and z generated by the aligner and $S(x) = \sum_{z \in \Gamma(x)} w(x, z)$. For embedding-based aligners we use embedding similarity as the score $w(x, z)$. If an aligner does not provide scores, we set all weights to 1.0.

Given a scored word alignment, we create a multilingual word alignment matrix W for each sentence as shown in Figure 2. Each cell contains 0 or 1 for Adamic-Adar or the alignment score for Weighted Adamic-Adar. We again apply Argmax to extract new alignment edges and then add them to the original alignment.

4 Experimental setup

4.1 PBC

The PBC corpus (Mayer and Cysouw, 2014) contains 1758 editions of the Bible in 1334 languages. The editions are aligned at the verse level and tokenized. A verse can contain more than one sentence, but we treat it as one unit in the parallel corpus since a true sentence level alignment is not available. There are some errors in tokenization (e.g., for Tibetan, Khmer and Chinese), but the overall quality of the corpus is good. For the majority of languages one edition is provided, while a few languages (in particular, English, French and German) contain up to dozens of editions. The verse coverage also differs from language to language. Some languages have translations of both New Testament and Hebrew Bible while others contain only one. Table 2 gives corpus statistics.

4.2 Word alignment datasets

PBC does not provide gold word alignments. To evaluate MPWA, we port two word alignment gold datasets of the Bible to PBC: Blinker (Melamed, 1998) and the recently published HELFI (Yli-Jyrä et al., 2020). We further experiment with bilingual datasets, using Machine Translation (MT) to create multiparallel corpora. Table 1 gives dataset statistics.

The HELFI dataset consists of the Greek New Testament, the Hebrew Bible and translations of both into Finnish. In addition, morpheme alignments are provided for Finnish-Greek and Finnish-Hebrew. We reformatted this dataset to the format used by PBC. In more detail, we added three new editions for the three languages to PBC. We identified the PBC verse identifier for each verse of HELFI to ensure proper verse alignment of these three new editions. The Finnish-Hebrew dataset has 22,291 verses and the Finnish-Greek dataset 7,909. We split these datasets 80/10/10 into train, validation and test.

The Blinker Bible dataset provides word level alignments of 250 Bible verses between English and French. The French side of this dataset matches with the edition Louis Segond 1910 in PBC. However, the tokenizations (Blinker vs PBC) are different. We therefore create a mapping of the tokens using character n-gram matching. For English, we created and added a new edition to PBC.

MT datasets. To more broadly evaluate MPWA, we also create multiparallel datasets for four non-Bible word alignment gold standards; these are listed in Table 1 as “Non-Bible” corpora. For these gold standards, we translate from English to all languages available in Google Translate, using their API.[4] For the added languages, we create alignments for the gold standard sentences using SimAlign.

4.3 Initial word alignments

We compare with two state of the art models, one statistical, one neural. Eflomal (Östling and Tiedemann, 2016) is a Bayesian statistical word aligner using Markov Chain Monte Carlo inference. SimAlign (Jalili Sabet et al., 2020) obtains word align-

[4] https://cloud.google.com/translate/docs/basic/translating-text

	Language Pair	Name	# Sentences (train/valid./test)
Bible	FIN-HEB	HELFI (Yli-Jyrä et al., 2020)	22291 (17832/2229/2230)
	FIN-GRC	HELFI (Yli-Jyrä et al., 2020)	7909 (6327/791/791)
	ENG-FRA	BLINKER (Melamed, 1998)	250
Non-Bible	ENG-DEU	EuroParl-based[a]	508
	ENG-FAS	(Tavakoli and Faili, 2014)	400
	ENG-HIN	WPT2005[b]	90
	ENG-RON	WPT2005[b]	203

[a] www-i6.informatik.rwth-aachen.de/goldAlignment/
[b] http://web.eecs.umich.edu/~mihalcea/wpt05/

Table 1: Overview of datasets. We use ISO 639-3 language codes. # Sentences: the number of available verses (i.e., sentences). FIN-HEB and FIN-GRC datasets split into train, validation and test.

# editions	1758
# languages	1334
# verses	20,470,892
# verses / # editions	11,520
# tokens / # verses	28.6

Table 2: PBC corpus statistics

ments from multilingual pretrained language models with no need for parallel data. For the symmetrization of Eflomal, we use grow-diag-final-and (GDFA) and intersection, and for SimAlign we use Argmax and Itermax. Intersection and Argmax generate accurate alignments while GDFA and Itermax are less accurate but have better coverage (Jalili Sabet et al., 2020).

We evaluate on a *target language pair* parallel sentence as follows: First, we create the matrix (Figure 2) for this sentence for all languages in the multiparallel corpus. Then we run link prediction on the matrix – this accumulates evidence from a set of languages in the multiparallel corpus. Finally, we take the predictions for the target language pair and add them to the original (bilingual) alignment.

NMF works best if it starts with high-accuracy (i.e., non-noisy) bilingual alignments – errors can result in incorrectly predicted alignment edges. We therefore use SimAlign Argmax and Eflomal Intersection, two word alignment methods with high precision, to create the initial alignments that are then fed into NMF. We then add the predictions to any desired original alignments; e.g., NMF (GDFA) uses Eflomal Intersection as the initial alignments and adds the predictions to Eflomal GDFA. See the Appendix for more details.

SimAlign offers high quality word alignments for well-represented languages from pretrained language models; however, our experiments show that its performance is far behind Eflomal for less well resourced languages like Biblical Hebrew and Koine Greek. Also, Eflomal is a better match for MPWA because it can provide word alignments for all languages available in a multiparallel corpus. In contrast, SimAlign is limited to languages supported by pretrained multilingual embeddings.

To feed Eflomal with enough training data for a target language pair, we use all available data from different translations of the language pair. For example if one language has two translations and the other one has three translations, Eflomal's training data will contain six aligned translation pairs for these two languages.

We use the standard evaluation measures for word alignment: precision, recall, F_1 and Alignment Error Rate (AER) (Och and Ney, 2003b; Östling and Tiedemann, 2016; Jalili Sabet et al., 2020).

5 Results

5.1 Multiparallel corpus results

We perform the first set of experiments on the Blinker Bible and the HELFI gold standards in the PBC. The baseline results are calculated on the original language pairs. MPWA can be applied to both Eflomal and SimAlign alignments. Since the default version of SimAlign can only generate alignments for the 84 languages that multilingual BERT supports,[5] for a better comparison, we use the same set of languages in the alignment graph for both SimAlign and Eflomal.

Table 3 shows the results for our methods applied on SimAlign and Eflomal baselines.[6] AdAd, NMF and WAdAd substantially improve the performance for all language pairs. SimAlign generates high-quality alignments for the English-French dataset, but cannot properly align underresourced languages like Biblical Hebrew and Koine Greek.

[5] https://github.com/google-research/bert/blob/master/multilingual.md

[6] We only consider SimAlign IterMax, not SimAlign ArgMax, because IterMax performed better throughout.

	Method	FIN-HEB				FIN-GRC				ENG-FRA			
		Prec.	Rec.	F_1	AER	Prec.	Rec.	F_1	AER	Prec.	Rec.	F_1	AER
Baseline	Eflomal (intersection)	**0.818**	0.269	0.405	0.595	**0.897**	0.506	0.647	0.353	**0.971**	0.521	0.678	0.261
	Eflomal (GDFA)	0.508	0.448	0.476	0.524	0.733	0.671	0.701	0.300	0.856	0.710	0.776	0.221
	SimAlign	0.190	0.113	0.142	0.858	0.366	0.265	0.307	0.693	0.886	0.692	0.777	0.221
Init SimAlign	AdAd	0.199	0.127	0.155	0.845	0.402	0.289	0.336	0.664	0.878	0.731	0.798	0.200
	WAdAd	0.186	0.165	0.175	0.825	0.353	0.350	0.351	0.649	0.856	0.752	0.801	0.197
	NMF	0.122	0.100	0.110	0.890	0.396	0.337	0.364	0.636	0.835	0.700	0.762	0.236
Init Eflomal	WAdAd (intersection)	0.781	0.612	**0.686**	**0.314**	0.849	0.696	**0.765**	**0.235**	0.938	0.689	0.794	0.203
	NMF (intersection)	0.78	0.576	0.663	0.337	0.864	0.669	0.754	0.248	0.948	0.624	0.753	0.245
	WAdAd (GDFA)	0.546	**0.693**	0.611	0.389	0.707	**0.783**	0.743	0.257	0.831	**0.796**	**0.813**	**0.186**
	NMF (GDFA)	0.548	0.646	0.593	0.407	0.72	0.759	0.739	0.261	0.844	0.767	0.804	0.195

Table 3: Comparison of results from different methods on PBC. The best results in each column are in bold. The three methods exploiting multiparallelism (AdAd, WAdAd, NMF) generally outperform the baselines on F_1 and AER.

In such cases, MPWA uses the accumulated information from all other language pairs in the graph to improve the performance. When starting with the SimAlign alignment ("Init SimAlign"), both methods improve the result for both FIN-HEB and FIN-GRC.

Eflomal generates better alignments for FIN-HEB and FIN-GRC. This means that Eflomal also generates better alignments between FIN, HEB and GRC on the one hand and the other languages in the graph on the other hand and therefore can provide a better signal for MPWA. The improvements of our models applied on Eflomal are higher than the ones applied on SimAlign for these language pairs.

When changing the initial alignments from Eflomal (intersection) to Eflomal (GDFA), we see different behaviors: GDFA improves the results for Blinker while it does not help for HELFI. We believe this is caused by the different ways the two datasets were annotated. In Blinker, many phrases are "exhaustively" aligned: if a phrase DE in English is aligned with FG in French then all four alignment edges (D-F, D-G, E-F, E-G) are given as gold edges.[7]

So Blinker contains a lot of many-to-many links. In contrast, most alignments are one-to-one in HELFI. This partially explains why intersection as initial alignment works much better for HELFI than GDFA and vice versa for Blinker.

In summary, compared to the baselines, we see very large improvements through exploiting multiparallelism for one type of alignment methodology (HELFI, F_1 improved by up to 20% for FIN-HEB) and improvements of up to 3.5% for the other (ENG-FRA).

5.2 MT dataset results

We perform the second set of experiments on gold standard alignments for language pairs that are not part of a multiparallel corpus such as PBC. To this end, we create artificial multiparallel corpora by translating the English side to all languages available in the Google Translate API. The main goal is to give broader evidence for the effectiveness of our method, beyond the specialized domain of the Bible.

Eflomal's alignments generally have good quality. However, they get worse when less parallel data is available (Jalili Sabet et al., 2020). Since the size of the multiparallel corpus created by machine translation is rather small, we use SimAlign for generating initial alignments. SimAlign has been shown to have good performance even for very small parallel corpora; in fact, it does not need any parallel data at all.

Table 4 shows the results of the experiments. Both NMF and WAdAd, improve the performance of the baseline by using the alignment graph. Improvements range from 0.8% (ENG-DEU) to 3.3% (ENG-HIN). This again demonstrates the utility of exploiting multiparallelism for word alignment. It is worth mentioning that in this case the translations are noisy since they were automatically generated. But even with these noisy translations (instead of a "true" multiparallel corpus), our models effectively leverage multiparallelism.

[7] For example, the alignment of the phrases "trembled violently" and "fut saisi d'und grande, d'une violente émotion" consists of 2 · 8 gold edges.

		ENG-PES				ENG-HIN				ENG-RON				ENG-DEU			
	Method	Prec.	Rec.	F_1	AER	Prec.	Rec.	F_1	AER	Prec.	Rec.	F_1	AER	Prec.	Rec.	F_1	AER
Baseline	SimAlign	0.756	0.645	0.696	0.304	0.709	0.493	0.582	0.418	0.807	0.663	0.728	0.272	0.829	0.795	0.812	0.188
Init SimAlign	AdAd	0.751	0.700	**0.725**	**0.276**	0.693	0.544	0.610	0.390	0.799	0.696	**0.744**	**0.256**	0.818	0.823	**0.820**	**0.179**
	WAdAd	0.705	**0.740**	0.722	0.278	0.643	**0.574**	0.607	0.394	0.725	**0.717**	0.721	0.279	0.749	**0.844**	0.794	0.207
	NMF	0.734	0.698	0.716	0.284	0.684	0.559	**0.615**	**0.385**	0.780	0.696	0.736	0.265	0.804	0.827	0.815	0.185

Table 4: Results with gold standards translated into other languages using machine translation. The best results in each column are in bold. The three methods exploiting multiparallelism (AdAd, WAdAd, NMF) outperform the baselines on F_1 and AER.

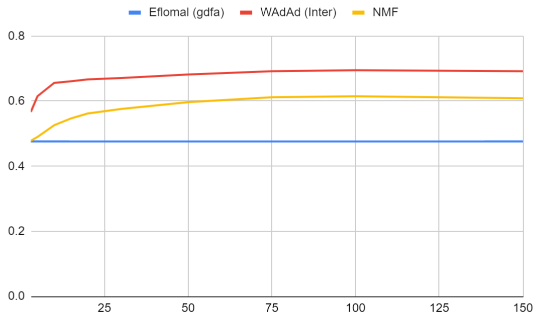


Figure 3: F_1 of MPWA for the target language pair FIN-HEB as a function of the number of additional languages. There is a clear rise initially. The curve flattens around 75.

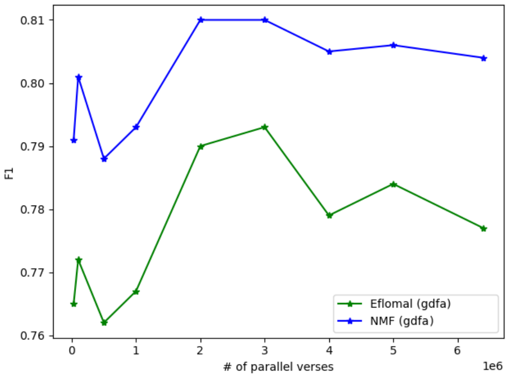


Figure 4: Word alignment F_1 on ENG-FRA as a function of the size of the training set, ranging from 30K to 6.4M training sentence pairs

5.3 Analysis

5.3.1 Effect of number of languages

The effect of adding more languages to the alignment graph is depicted in Figure 3. This plot shows F_1 for FIN-HEB. As seen in the figure, the slope is pretty steep up to 25 languages, but even adding just three languages can still improve the results. For 75 languages we have almost reached the peak and after 100, adding more languages is not improving the results. This means that MPWA can also be helpful for corpora with a smaller number of languages – a massively parallel corpus with thousands of languages is not required.

5.3.2 Size of the training set

To assess the effect of dataset size on the performance of MPWA, we perform a set of experiments on ENG-FRA and NMF. To this end, we take the training data for ENG-FRA and train models on subsets of it. The training data consists of 6.4M sentence pairs – this number is so high because we use the crossproduct of all editions in English and French (§4.3).

The results are shown in Figure 4. Eflomal performance increases with training set size initially and is then less predictable. NMF consistently improves the scores.

5.3.3 Effect of task difficulty

Table 3 shows large improvements for all datasets, and especially for FIN-HEB and FIN-GRC. To get more insight into the reasons for this improvement, we stratify FIN-HEB verses by dividing the interval $[0, 1]$ of initial F_1 performance of Eflomal into five equal-sized subintervals: $[0, 0.2], \dots, (0.8, 1]$.

Figure 5 indicates that MPWA is most effective for difficult verses, but brings little improvement for easy verses. We attribute this to two reasons:

1. An easy to align verse in a language pair cannot use help from other languages since it already has good alignment links (although the language pair would still be of benefit in improving alignments for the sentence in other languages). So there is no way for MPWA to get better results in this scenario.

2. MPWA only tries to get better results by adding new alignments, and as an easy verse already has many alignment links, adding new links almost inevitably results in a drop in pre-

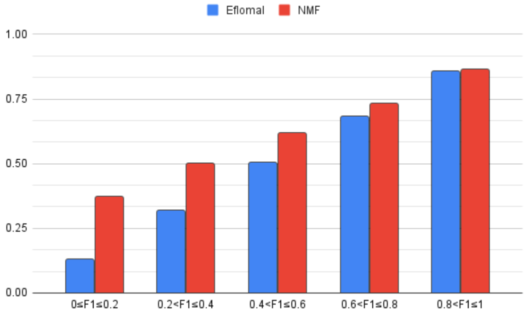


Figure 5: How helpful is MPWA for different difficulty levels? For this analysis, FIN-HEB verses were stratified according to Eflomal F_1 (x-axis). We see that MPWA brings the largest improvements for difficult sentences.

ENG-FRA		FIN-HEB		FIN-GRC	
Lang.	Δ	Lang.	Δ	Lang.	Δ
SPA	+2.0%	TGL	+17.7%	LAT	+7.5%
ITA	+1.9%	FRY,ELL	+17.3%	ELL	+6.6%
DEU	+1.8%	SWE	+17.3%	ENG	+6.1%
NLD	+1.4%	NLD	+16.8%	FRY	+5.8%
AFR	+1.3%	YOR	+14.2%	BEL	+5.7%

Table 5: The five most helpful languages and WAdAd's absolute improvements in F_1 over the initial bilingual aligner SimAlign. For example, MPWA improves the bilingual FIN-GRC alignment by 7.5% if applied to the trilingual corpus FIN-GRC-LAT, i.e., Latin can be viewed as the best bridge between Finnish and Greek.

cision. It may also be possible to inspect and prune existing Eflomal links using MPWA to get better results in this scenario.

5.3.4 Most helpful languages

For each dataset, the five most helpful languages with their corresponding improvements are listed in Table 5. We hypothesize that these languages serve to bridge the typological gap between the two target languages. Table 5 suggests one should be able to achieve excellent results – even for a corpus with a small number of languages – if we utilize an intelligent selection of languages.

5.3.5 Multiple translations in two languages

There are some datasets that contain few languages, but many translations of a text in one language. PBC is one example of such a dataset, many literary works another (e.g., many novels have many translations in English). To see whether MPWA can also help in this scenario, we picked all available 49 English and French editions from PBC and used them as additional translations for the ENG-FRA dataset. The outcome of this experiment is compared with the outcome of the same setup, but with translations from languages other than French and English in Table 6. From this table we can conclude that translations from the target language pair can also assist, but not as much as translations from other languages.

	Prec.	Rec.	F_1	AER
Eflomal (intersection)	0.971	0.521	0.678	0.319
Eflomal (GDFA)	0.856	0.710	0.776	0.221
NMF (target languages)	0.830	0.749	0.787	0.213
NMF (other languages)	0.837	0.753	0.793	0.205

Table 6: F_1 for ENG-FRA. MPWA can exploit a multiparallel corpus with languages different from the target languages ("other languages") better than one that contains additional translations in the target languages ("target languages").

6 Conclusion and Future Work

We presented MPWA, a framework for leveraging multiparallel corpora for word alignment. We used two prediction methods, one based on recommender systems and one based on link prediction algorithms. By adding new alignment edges to the output of bilingual aligners, both methods show large improvements over the bilingual baselines, with absolute improvements of F_1 of up to 20%. We have also ported Blinker and HELFI word alignment gold standards to the Parallel Bible Corpus in the hope that this will help foster more work on exploiting multiparallel corproa.

Future work. In this paper, we have mainly focused on *adding* new alignment edges to baseline word alignments based on properties of the multiparallel alignment graph. This increases recall, but can harm precision. In future work, we plan to expand on the possibility of *deleting* edges based on evidence from the multiparallel alignment graph (cf. 5.3.3), thereby potentially improving both precision and recall.

Acknowledgments

This work was supported by the European Research Council (ERC, Grant No. 740516) and the German Federal Ministry of Education and Research (BMBF, Grant No. 01IS18036A). The fourth author was supported by the Bavarian research institute for digital transformation (bidt) through their fellowship program. We thank the anonymous reviewers for their constructive comments.

References

Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social networks*, 25(3):211–230.

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 54–65, Berlin, Germany. Association for Computational Linguistics.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, pages 108–117, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science*, 286(5439):509–512.

Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, and Roldano Cattoni. 2008. Phrase-based statistical machine translation with pivot languages. In *International Workshop on Spoken Language Translation (IWSLT) 2008*.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

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).

Jaegul Choo, Changhyun Lee, Chandan K Reddy, and Haesun Park. 2013. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE transactions on visualization and computer graphics*, 19(12):1992–2001.

Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735, Prague, Czech Republic. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

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.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1520–1530, Melbourne, Australia. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Karim Filali and Jeff Bilmes. 2005. Leveraging multiple languages to improve statistical MT word alignments. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 92–97. IEEE.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Ayyoob ImaniGooghari, Masoud Jalili Sabet, Philipp Dufter, Michael Cysou, and Hinrich Schütze. 2021. ParCourE: A parallel corpus explorer for a massively multilingual corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 63–72, Online. 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.

Glen Jeh and Jennifer Widom. 2002. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Da Kuang, Jaegul Choo, and Haesun Park. 2015. Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional Clustering Algorithms*, pages 215–243. Springer.

Shankar Kumar, Franz J. Och, and Wolfgang Macherey. 2007. Improving word alignment with bridge languages. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 42–50, Prague, Czech Republic. Association for Computational Linguistics.

Adrien Lardilleux and Yves Lepage. 2008a. A truly multilingual, high coverage, accurate, yet simple, sub-sentential alignment method. In *The 8th conference of the Association for Machine Translation in the Americas (AMTA 2008)*, pages 125–132, Waikiki, Honolulu, United States.

Adrien Lardilleux and Yves Lepage. 2008b. Multilingual alignments by monolingual string differences. In *Coling 2008: Companion volume: Posters*, pages 55–58, Manchester, UK. Coling 2008 Organizing Committee.

Adrien Lardilleux and Yves Lepage. 2009. Sampling-based multilingual alignment. In *Proceedings of the International Conference RANLP-2009*, pages 214–218, Borovets, Bulgaria. Association for Computational Linguistics.

William D. Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Linyuan Lü and Tao Zhou. 2010. Link prediction in weighted networks: The role of weak ties. *EPL (Europhysics Letters)*, 89(1):18001.

Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. 2014. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284.

Kelly Marchisio, Conghao Xiong, and Philipp Koehn. 2021. Embedding-enhanced GIZA++: Improving alignment in low-and high-resource scenarios using embedding space geometry. *arXiv preprint arXiv:2104.08721*.

Thomas Mayer and Michael Cysouw. 2012. Language comparison through sparse multilingual word alignment. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 54–62, Avignon, France. 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).

I. Dan Melamed. 1998. Manual annotation of translational equivalence: The blinker project. *CoRR*, cmp-lg/9805005.

Mathias Müller. 2017. Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46, Copenhagen, Denmark. Association for Computational Linguistics.

Preslav Nakov and Hwee Tou Ng. 2012. Improving statistical machine translation for a resource-poor language using related resource-rich languages. *Journal of Artificial Intelligence Research*, 44:179–222.

Franz Josef Och and Hermann Ney. 2003a. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Franz Josef Och and Hermann Ney. 2003b. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Robert Östling. 2014. Bayesian word alignment for massively parallel texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 123–127. The Association for Computer Linguistics.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, Beijing, China. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1).

Mohammad Sadegh Rasooli, Noura Farra, Axinia Radeva, Tao Yu, and Kathleen McKeown. 2018. Cross-lingual sentiment transfer with limited resources. *Machine Translation*, 32(1):143–165.

Sascha Rothe and Hinrich Schütze. 2014. CoSimRank: A flexible & efficient graph-theoretic similarity measure. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1392–1402, Baltimore, Maryland. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons. 2006. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021. Bilingual lexicon induction via unsupervised bitext construction and word alignment. 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 813–826, Online. Association for Computational Linguistics.

Leila Tavakoli and Heshaam Faili. 2014. Phrase alignments in parallel corpus using bootstrapping approach. *International Journal of Information & Communication Technology Research*, 6(3).

Shijie Wu and Mark Dredze. 2020. Do explicit alignments robustly improve multilingual encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482, Online. Association for Computational Linguistics.

Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273.

Anssi Yli-Jyrä, Josi Purhonen, Matti Liljeqvist, Arto Antturi, Pekka Nieminen, Kari M. Räntilä, and Valtter Luoto. 2020. HELFI: a Hebrew-Greek-Finnish parallel Bible corpus with cross-lingual morpheme alignment. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4229–4236, Marseille, France. European Language Resources Association.

Hamed Zamani, Javid Dadashkarimi, Azadeh Shakery, and W Bruce Croft. 2016. Pseudo-relevance feedback based on matrix factorization. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1483–1492.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems*, 31:5165–5175.

Shaojun Zhao and Daniel Gildea. 2010. A fast fertility hidden Markov model for word alignment using MCMC. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 596–605, Cambridge, MA. Association for Computational Linguistics.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630.

A Pipeline Details

There are several elements of the MPWA pipeline that can be configured by the user, e.g., depending on whether precision or recall are more important for an application. Here we show in Figures 6 and 7 the two pipeline configurations we used for the results in the paper.

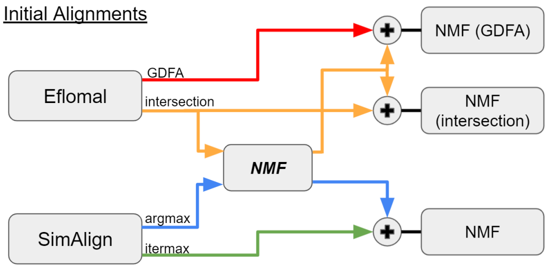


Figure 6: The pipeline for NMF alignments

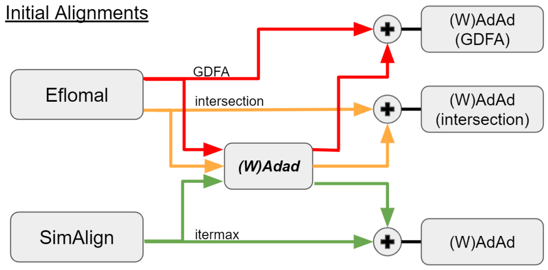


Figure 7: The pipeline for AdAd and WAdAd alignments

Chapter 4

Graph Neural Networks for Multiparallel Word Alignment

Graph Neural Networks for Multiparallel Word Alignment

Ayyoob Imani[1], **Lütfi Kerem Şenel**[1], **Masoud Jalili Sabet**[1],
François Yvon[2], **Hinrich Schütze**[1]
[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2]Université Paris-Saclay, CNRS, LISN, France
`{ayyoob, masoud, lksenel}@cis.lmu.de,`
`francois.yvon@limsi.fr`

Abstract

After a period of decrease, interest in word alignments is increasing again for their usefulness in domains such as typological research, cross-lingual annotation projection and machine translation. Generally, alignment algorithms only use bitext and do not make use of the fact that many parallel corpora are multiparallel. Here, we compute high-quality word alignments between multiple language pairs by considering all language pairs together. First, we create a multiparallel word alignment graph, joining all bilingual word alignment pairs in one graph. Next, we use graph neural networks (GNNs) to exploit the graph structure. Our GNN approach (i) utilizes information about the meaning, position and language of the input words, (ii) incorporates information from multiple parallel sentences, (iii) adds and removes edges from the initial alignments, and (iv) yields a prediction model that can generalize beyond the training sentences. We show that community detection provides valuable information for multiparallel word alignment. Our method outperforms previous work on three word alignment datasets and on a downstream task.

1 Introduction

Word alignments are crucial for statistical machine translation (Koehn et al., 2003) and useful for many other multilingual tasks such as neural machine translation (Alkhouli and Ney, 2017; Alkhouli et al., 2016), typological analysis (Lewis and Xia, 2008; Östling, 2015; Asgari and Schütze, 2017) and annotation projection (Yarowsky and Ngai, 2001; Fossum and Abney, 2005; Wisniewski et al., 2014; Huck et al., 2019). The rise of deep learning initially led to a temporary plateau, but interest in word alignments is now increasing, demonstrated by several recent publications (Jalili Sabet et al., 2020; Chen et al., 2020; Dou and Neubig, 2021).

While word alignment is usually considered for bilingual corpora, our work addresses the problem of *word alignment in multiparallel corpora*, containing sentence level parallel text in more than two languages. Examples of multiparallel corpora are JW300 (Agić and Vulić, 2019), PBC (Mayer and Cysouw, 2014) which covers the highest number of languages (1334), and Tatoeba.[1] While the per-language amount of data provided in such corpora is less than bilingual corpora, they support highly low-resource languages, many of which are not covered by existing language technologies (Joshi et al., 2020). Therefore, these corpora are essential for studying many of the world's low-resource languages.

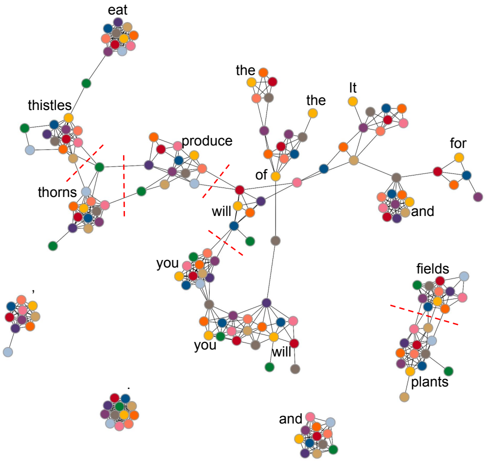


Figure 1: Alignment graph for the verse "It will produce thorns and thistles for you, and you will eat the plants of the field." in a 12-way multiparallel corpus. Colors represent languages. Each English (yellow) node is annotated with its word. Red dashed lines cut links that incorrectly connect distinct concepts. We exploit community detection algorithms to detect distinct concepts. This provides valuable information for our GNN model and improves word alignments.

We consider the task of word alignment for multiparallel sentences. The basic motivation is that the alignment between words in languages U and

[1] https://tatoeba.org

V can benefit from word-level alignments of U and V with a translation in a third language W. Following up on the work of Imani Googhari et al. (2021), we model multilingual word alignments with tools borrowed from graph theory (community detection algorithms) combined with neural network based models, specifically, the graph neural network (GNN) model of Scarselli et al. (2009).

GNNs were proposed to extend the powerful current generation of neural network models to the processing of graph-structured data and they have gained increasing popularity in many domains (Wu et al., 2020; Sanchez-Gonzalez et al., 2018; He et al., 2020). GNNs can incorporate heterogeneous sources of signal in the form of node and edge features. We use this property to take into account properties of the whole alignment graph, notably its tendency to cluster into *communities*, see Figure 1.

With our new proposed methods, we obtain improved results on word alignment for three language pairs: English-French, Finnish-Hebrew and Finnish-Greek. As a demonstration of the importance of high-quality alignments, we use our word alignments to project annotations from high-resource to low-resource languages. We improve a part-of-speech tagger for Yoruba by training it over a high-quality dataset, which is created using annotation projection.

Contributions: **i)** We propose a graph neural network model to enhance word alignments in a multiparallel corpus. The model incorporates a diverse set of features for word alignments in multiparallel corpora and an elegant way of training it efficiently and effectively. **ii)** We show that community detection improves multiparallel word alignment. **iii)** We show that the improved alignments improve performance on a downstream task for a low resource language. **iv)** We propose a new method to infer alignments from the alignment probability matrix. **v)** We will make our code publicly available.

2 MultiParallel Word Alignment Graphs

2.1 Building MultiParallel Word Alignment Graphs

Our starting point is the work of Imani Googhari et al. (2021), who introduce MPWA (MultiParallel Word Alignment), a framework that utilizes the synergy between multiple language pairs to improve bilingual word alignments for a target language pair. The rationale is that some of the missing alignment edges between a source and a target language can be recovered using their alignments with words in other languages.

An MPWA graph is constructed using the following two steps:

1. create initial bilingual alignments for all language pairs in a multiparallel corpus using a bilingual word aligner;

2. represent the bilingual alignments for each multiparallel sentence in a graph containing one vertex for each token occurring in any language and one edge for each initial bilingual word alignment link.

Potentially missing alignment links are then added based on the graph structure in an inference step, casting the word alignment task as an edge prediction problem. Figure 1 gives an example of a multiparallel word alignment graph for a 12-way multiparallel sentence.

Imani Googhari et al. (2021) use two traditional graph algorithms, Adamic-Adar and non-negative matrix factorization, for predicting new alignment edges from the MPWA graph. However, these graph algorithms are applied to individual multiparallel sentences independently and therefore cannot accumulate knowledge from multiple sentences. Moreover, their edge predictions are solely based on the structure of the graph and do not take advantage of other beneficial signals such as a word's language, relative position and meaning. Another limitation of this work is that it only adds links and does not remove any, which is important to improve precision.

This work addresses these shortcomings by using GNNs to predict alignment edges from MPWA graphs.

2.2 Community Detection in Alignment Graphs

One important advantage of GNNs over traditional graph algorithms is that they can directly incorporate signals from different sources in the form of node and edge features. We utilize this by taking into account the following observation: The nodes in the alignment graph are words in parallel sentences that are translations of each other. If the initial bilingual alignments are of good quality, we expect words that are mutual translations to form densely connected regions or *communities*; see Figure 1. These communities should not be linked to each other, each corresponding to a distinct connected component. In other words, ideally, words representing a concept should be densely connected, but there should be no links between different concepts. While, this intuition will not be true for all concepts between all possible language pairs, we nonetheless hypothesize that identifying distinct concepts in a multiparallel word alignment graph can provide useful information.

To examine to what extent these expectations are met, we count the components in the original Eflomal-generated (Östling and Tiedemann, 2016) graph (see §4.2 for details on the initial alignments). Table 1 shows that the average number of components per sentence is less than three ("Eflomal intersection", columns #CC). But intuitively, the number of components should roughly correspond to sentence length (or, more precisely, the number of content words). This indicates that there are many links that incorrectly connect different concepts. To detect such links, we use community detection (CD) algorithms.

CD algorithms find subnetworks of nodes that form tightly knit groups that are only loosely connected with a small number of links (Girvan and Newman, 2002). One well-known approach to CD attempts to maximize the modularity measure (Newman and Girvan, 2004). Modularity assesses how beneficial a division of a community into two communities is, in the sense that there are many links within communities and only a few between them. Given a graph G with n nodes and m edges and G's adjacency matrix $A \in \mathbb{R}^{n \times n}$, modularity is defined as:

$$mod = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \gamma \frac{d_i d_j}{2m} \right) I(c_i, c_j) \quad (1)$$

where d_i is the degree of node i. $I(c_i, c_j)$ is 1 if nodes i and j are in the same community, 0 otherwise.

As exact modularity maximization is intractable, we experiment with two CD algorithms implementing different heuristic approaches:

- Greedy modularity communities (GMC). This method uses Clauset-Newman-Moore greedy modularity maximization (Clauset et al., 2004). GMC begins with each node in its own community and greedily joins the pair of communities that most increases modularity until no such pair exists.

	FIN-HEB		FIN-GRC		ENG-FRA	
	#CC	F_1	#CC	F_1	#CC	F_1
Eflomal intersection	2.2	0.404	1.6	0.646	2.2	0.678
GMC	13.7	0.396	10.1	0.375	13.5	0.411
LPC	41.5	0.713	37.1	0.754	46.0	0.767
Sentence length	25.7		23.2		27.4	

Table 1: Effect of community detection algorithms (GMC and LPC) on alignment prediction. #CC: average number of connected components. F_1: word alignment performance.

- Label propagation communities (LPC). This method finds communities in a graph using label propagation (Cordasco and Gargano, 2010). It begins by giving a label to each node of the network. Then each node's label is updated by the most frequent label among its neighbors in each iteration. It performs label propagation on a portion of nodes at each step and quickly converges to a stable labeling.

After detecting communities, we link all nodes inside a community and remove all inter-community links. GMC (LPC) on average removes 3% (7%) of the edges. Table 1 reports the average number of graph components per sentence before and after running GMC and LPC, as well as the corresponding F_1 for word alignment (see §4.1 for details on the evaluation datasets). We see that the number of communities found is lower for GMC than for LPC; therefore, LPC identifies more candidate links for deletion.[2] Comparing the number of communities detected with the average sentence length, GMC seems to have failed to detect enough communities to split different concepts properly. The F_1 scores confirm this observation and show that LPC performs well at detecting the communities we are looking for.

This analysis shows that CD algorithms compute valuable information for word alignments. To exploit this in our GNN model, we add node community information as a node feature; see §3.1.3.

3 Predicting and using MultiParallel Word Alignments (MPWAs)

3.1 GNNs for MPWA

GNNs can be used in transductive or inductive settings. Transductively, the final model can only be

[2] LPC may detect more communities than average sentence length because of null words: words that have no translation in the other languages, giving rise to separate communities.

used for inference over the same graph that it is trained on. In an inductive setting, which we use here, nodes are represented as feature vectors, and the final model has the advantage of being applicable to a different graph in inference.

Below is the step-by-step overview of our GNN-based approach for an MPWA graph:

1. run community detection algorithms on the initial graph (§2.2);

2. obtain features for the nodes of the graph (§3.1.3);

3. compute node embeddings from node features and initial alignment links in the GNN encoding step (§3.1.2);

4. learn to distinguish between nodes that are aligned together and that are not aligned together in the GNN decoding step (§3.1.2);

After the GNN model is trained on multiple MPWA graphs, it is used to infer an alignment probability matrix between tokens in a source language and tokens in a target language for a multiparallel sentence, see §3.1.4. Our method predicts new alignment links from this matrix, independently of initial edges. Therefore, given an initial bilingual alignment, it is not limited to adding edges, but it can also remove them.

3.1.1 Model Architecture

Our model is inspired by the Graph Auto Encoder (GAE) model of Kipf and Welling (2016) for link prediction. A GAE model consists of an encoder and a decoder. The encoder creates a hidden representation for each node of the graph and the decoder predicts the links of the graph given the nodes' representations. Using the graph of word alignments, the model will learn the word alignment task. Therefore it will be able to predict word alignments that are missed by the original bilingual word aligner and also detect incorrect alignment edges.

We make changes to this model to improve the model's quality and reduce its computational cost. We use GATConv layers (Veličković et al., 2018) for the encoder instead of GCNConv (Kipf and Welling, 2017) and a more sophisticated decoder instead of simple dot product for a stronger model. We also introduce a more efficient training procedure.

The **encoder** is a graph attention network (GAT) (Veličković et al., 2018) with two GATConv layers followed by a fully connected layer. Layers are connected by RELU non-linearities. A GATConv layer computes its output $\mathbf{x}'_i$ for a node i from its input $\mathbf{x}_i$ as

$$\mathbf{x}'_i = \alpha_{i,i}\mathbf{W}\mathbf{x}_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\mathbf{W}\mathbf{x}_j, \qquad (2)$$

where $\mathbf{W}$ is a weight matrix, $\mathcal{N}(i)$ is some *neighborhood* of node i in the graph, and $\alpha_{i,j}$ is the attention coefficient indicating the importance of node j's features to node i. $\alpha_{i,j}$ is computed as

$$\alpha_{i,j} = \frac{\exp\left(\mathrm{g}\left(\mathbf{a}^\top[\mathbf{W}\mathbf{x}_i \,\|\, \mathbf{W}\mathbf{x}_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)\cup\{i\}} \exp\left(\mathrm{g}\left(\mathbf{a}^\top[\mathbf{W}\mathbf{x}_i \,\|\, \mathbf{W}\mathbf{x}_k]\right)\right)} \qquad (3)$$

where $\|$ is concatenation, g is LeakyReLU, and $\mathbf{a}$ is a weight vector. Given the features for the nodes and their alignment edges, the encoder creates a contextualized hidden representation for each node.

Based on the hidden representations of two nodes, the **decoder** predicts whether a link connects them. The decoder architecture consists of a fully connected layer, a RELU non-linearity and a sigmoid layer.

3.1.2 Training

By default, GAE models are trained using full batches with random negative samples. This approach requires at least tens of epochs over the training dataset to converge and a lot of GPU memory for graphs as large as ours. We train our model using mini-batches to decrease memory requirements and improve the performance. Using our training approach the model converges after one epoch. We take care to select informative negative samples (as opposed to random selection) as described below.

Figure 2 displays our GNN model and the training process. The training set contains one graph for each sentence. Each graph is divided into multiple batches. Each batch contains a random subset of the graph's edges as positive samples. The negative samples are created as follows. Given a sentence $u_1u_2\ldots u_n$ in language U and its translation $v_1v_2\ldots v_m$ in language V, for each alignment edge u_i:v_j in the current batch, two negative edges u_i:v'_j and u'_i:v_j ($j' \neq j$, $i' \neq i$) are randomly sampled.

For each training batch, the encoder takes the batch's whole graph (i.e., node features for all

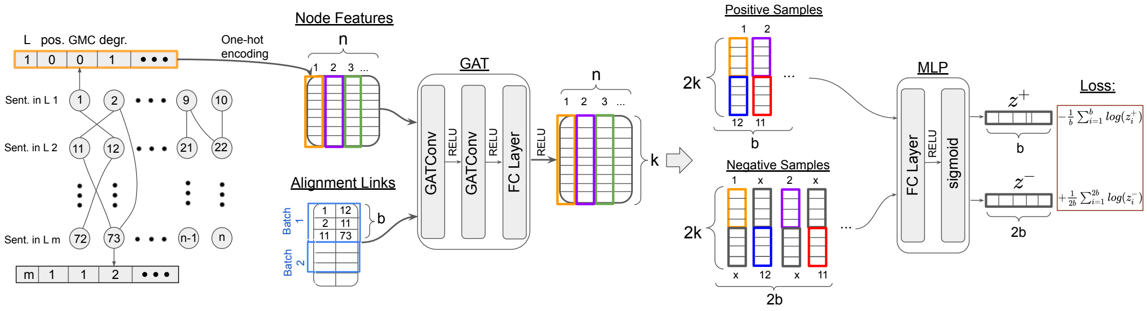


Figure 2: GNN training. At each training step, node features and links of a multiparallel sentence are fed to a graph attention network (GAT) that creates hidden representations for all nodes. On the decoder side, at each step, one batch of alignment links and hidden node representations is used to create positive and negative samples, which are then processed and classified by a multi-layer perceptron (MLP). Parameters of GAT and MLP are updated for each batch. FC = fully connected.

graph nodes and all graph edges) as input and computes hidden representations for the nodes. On the decoder side, for each link between two nodes in the batch, the hidden representations of the two nodes are concatenated to create the decoder's input. The decoder's target is the link class: 1 (resp. 0) for positive (resp. negative) links. We train with a binary classification objective:

$$\mathcal{L} = -\frac{1}{b}\sum_{i=1}^{b}\log(p_i^+) + \frac{1}{2b}\sum_{i=1}^{2b}\log(p_i^-) \qquad (4)$$

where b is the batch size and p_i^+ and p_i^- are the model predictions for the i^{th} positive and negative samples within the batch. Parameters of the encoder and decoder as well as the node-embedding feature layer are updated after each training step.

3.1.3 Node Features

We use three main types of node features: (i) graph structural features, (ii) community-based features and (iii) word content features.

Graph structural features. We use *degree, closeness* (Freeman, 1978) , *betweenness* (Brandes, 2001) , *load* (Newman, 2001) and *harmonic centrality* (Boldi and Vigna, 2014) features as additional information about the graph structure. These features are continuous numbers, providing information about the position and connectivity of the nodes within the graph. We standardize (i.e., z-score) each feature across all nodes, and train an embedding of size four for each feature.[3]

Community-based features. One way to incorporate community information into our model is to train the model based on a refined set of edges after the community detection step. This approach hobbles the GNN model by making decisions about many of the edges before the GNN gets to see them. Our initial experiments also confirmed that training the GNN over CD refined edges does not help. Therefore, we add community information as node features and let the GNN use them to improve its decisions. We use the community detection algorithms GMC and LPC (see §§2.2) to identify communities in the graph. Then we represent the community membership information of the nodes as one-hot vectors and learn an embedding of size 32 for each of the two algorithms.

Word content features. We train embeddings for *word position* (size 32) and *word language* (size 20). We learn 100-dimensional multilingual *word embeddings* using Levy et al. (2017)'s sentence-ID method on the 84 PBC languages selected by Imani Googhari et al. (2021). Word embeddings serve as initialization and are updated during GNN training.

After concatenating these features, each node is represented by a 236 dimensional vector that is then fed to the encoder.

3.1.4 Inducing Bilingual Alignment Edges

Given a source sentence $\hat{x} = x_1, x_2, \ldots, x_m$ in language X and a target sentence $\hat{y} = y_1, y_2, \ldots, y_l$ in language Y, we feed all possible alignment links between source and target to the decoder. This produces a symmetric alignment probability matrix S of size $m \times l$ where S_{ij} is the predicted alignment probability between words x_i and y_j. Using these values directly to infer alignment edges is usually suboptimal; therefore, more sophisticated methods

[3] Learning a size-four embedding instead of a single number gives the feature a weight similar to other features – which have a feature vector of about the same size.

have been suggested (Ayan and Dorr, 2006; Liang et al., 2006). Here we propose a new approach: it combines Koehn et al. (2005)'s Grow-Diag-Final-And (GDFA) with Dou and Neubig (2021)'s probability thresholding. We modify the latter to account for the variable size of the probability matrix (i.e., length of source/target sentences). Our method is not limited to adding new edges to some initial bilingual alignments, a limitation of prior work. As we predict each edge independently, some initial links can be discarded from the final alignment.

We start by creating a set of *forward* (source-to-target) alignment edges and a set of *backward* (target-to-source) alignment edges. To this end, first, inspired by probability thresholding (Dou and Neubig, 2021), we apply softmax to S, and zero out probabilities below a threshold to get a source-to-target probability matrix S^{XY}:

$$S^{XY} = S * (\text{softmax}(S) > \frac{\alpha}{l}) \tag{5}$$

Analogously, we compute the target-to-source probability matrix S^{YX}:

$$S^{YX} = S^{\top} * (\text{softmax}(S^{\top}) > \frac{\alpha}{m}) \tag{6}$$

where α is a sensitivity hyperparameter, e.g., $\alpha = 1$ means that we pick edges with a probability higher than average. We experimentally set $\alpha = 2$. Next, from each row of S^{XY} (S^{YX}), we pick the cell with the highest value (if any exists) and add this edge to the *forward* (*backward*) set.

We create the final set of alignment edges by applying the GDFA symmetrization method (Koehn et al., 2005) to *forward* and *backward* sets. The gist of GDFA is to use the intersection of *forward* and *backward* as initial alignment edges and add more edges from the union of *forward* and *backward* based on a number of heuristics. We call this method *TGDFA* (Thresholding GDFA).

We also experiment with combining TGDFA with the original bilingual GDFA alignments. We do so by adding bilingual GDFA edges to the union of *forward* and *backward* before performing the GDFA heuristics. We refer to these alignments as *TGDFA+orig*.

We evaluate the resulting alignments using F_1 score and alignment error rate (AER), the standard metrics in the word alignment literature.

3.2 Annotation Projection

Annotation projection automatically creates linguistically annotated corpora for low-resource languages. A model trained on data with "annotation-projected" labels can perform better than a completely unsupervised method. Here, we focus on universal part-of-speech (UPOS) tagging (Petrov et al., 2012) for the low resource target language Yoruba; this language only has a small set of annotated sentences in Universal Dependencies (Nivre et al., 2020) and has poor POS results in unsupervised settings (Kondratyuk and Straka, 2019).

The quality of the target annotated corpus depends on the quality of the annotations in the source languages and the quality of the word alignments between sources and target. We use the Flair (Akbik et al., 2019) POS taggers for three high resource languages, English, German and French (Akbik et al., 2018), to annotate 30K verses whose Yoruba translations are available in PBC. We then transfer the POS tags from source to target using three different approaches: (i) We directly transfer annotations from English to the target. (ii) For each word in the target, we get its Eflomal bilingual alignments in the three source languages and predict the majority POS to annotate the target word. (iii) The same as in (ii), but we use our GNN (TGDFA) alignments (instead of Eflomal alignments) to project from source to target. In all three approaches, we discard any target sentence from the POS tagger training data if more than 50% of its words are annotated with the "X" (other) tag.

We train a Flair SequenceTagger model on the target annotated data using mBERT embeddings (Devlin et al., 2019) and evaluate on Yoruba test from Universal Dependencies.[4]

4 Experimental Setup

4.1 Word Alignment Datasets

Following Imani Googhari et al. (2021), we use PBC, a multiparallel corpus of 1758 sentence-aligned editions of the Bible in 1334 languages.

Evaluation data. For our main evaluation, we use the two word alignment gold datasets for PBC published by Imani Googhari et al. (2021): Blinker (Melamed, 1998) and HELFI (Yli-Jyrä et al., 2020). **The HELFI dataset** contains the Hebrew Bible, Greek New Testament and their translations into Finnish. For HELFI, we use Imani Googhari et al. (2021)'s train/dev/test splits. **The Blinker dataset** provides word level alignments between English and French for 250 Bible verses.

[4] https://universaldependencies.org/

	FIN-HEB				FIN-GRC				ENG-FRA			
Method	Prec.	Rec.	F_1	AER	Prec.	Rec.	F_1	AER	Prec.	Rec.	F_1	AER
Eflomal (intersection)	**0.818**	0.269	0.405	0.595	**0.897**	0.506	0.647	0.353	**0.971**	0.521	0.678	0.261
Eflomal (GDFA)	0.508	0.448	0.476	0.524	0.733	0.671	0.701	0.300	0.856	0.710	0.776	0.221
WAdAd (intersection)	0.781	0.612	0.686	0.314	0.849	0.696	0.765	0.235	0.938	0.689	0.794	0.203
NMF (intersection)	0.780	0.576	0.663	0.337	0.864	0.669	0.754	0.248	0.948	0.624	0.753	0.245
WAdAd (GDFA)	0.546	**0.693**	0.611	0.389	0.707	**0.783**	0.743	0.257	0.831	**0.796**	0.813	0.186
NMF (GDFA)	0.548	0.646	0.593	0.407	0.720	0.759	0.739	0.261	0.844	0.767	0.804	0.195
GNN (TGDFA)	0.811	0.648	**0.720**	**0.280**	0.845	0.724	**0.780**	**0.220**	0.926	0.711	0.804	0.192
GNN (TGDFA+orig)	0.622	0.683	0.651	0.349	0.738	0.780	0.758	0.242	0.863	0.789	**0.824**	**0.174**

Table 2: Word alignment results on PBC for GNN and baselines. The best result in each column is in bold. GNN outperforms the baselines as well as the graph algorithms WAdAd and NMF on F_1 and AER.

Training data. The graph algorithms used by Imani Googhari et al. (2021) operate on each multiparallel sentence separately. In contrast, our approach allows for an inductive setting where a model is trained on a training set, accumulating knowledge from multiple multiparallel sentences. We combine the verses in the training sets of Finnish-Hebrew and Finnish-Greek for a combined training set size of 24,159.[5]

4.2 Initial Word Alignments

We use the Eflomal statistical word aligner to obtain bilingual alignments. We train it for every language pair in our experiments. We do not consider SimAlign (Jalili Sabet et al., 2020) since it is shown to perform poorly for languages whose representations in the multilingual pretrained language model are of low quality. We use Eflomal asymmetrical alignments post-processed with the intersection heuristic to get high precision bilingual alignments as input to the GNN. We use the same subset of 84 languages as Imani Googhari et al. (2021).

4.3 Training Details

We use PyTorch Geometric[6] to construct and train the GNN. The model's hidden layer size is 512 for both GATConv and Linear layers. We train for one epoch on the training set – a small portion of the training set is enough to learn good embeddings (see §5.1.1). For training, we use a batch size of 400 and learning rate of .001 with AdamW (Loshchilov and Hutter, 2017). The whole training process takes less than 4 hours on a GeForce GTX 1080 Ti and the inference time is on the order of milliseconds per sentence.

5 Experiments and Results

5.1 Multiparallel corpus results

Table 2 shows results on Blinker and HELFI for our GNNs and the baselines: bilingual alignments and two graph-based algorithms WAdAd and NMF from Imani Googhari et al. (2021). Our GNNs yield a better trade-off between precision and recall, most likely thanks to their ability to remove edges, and achieve the best F_1 and AER on all three datasets, outperforming WAdAd and NMF.

GNN (TGDFA) achieves the best results on HELFI (FIN-HEB, FIN-GRC) while GNN (TGDFA+orig) is best on Blinker (ENG-FRA). As argued in Imani Googhari et al. (2021), this is mostly due to the different ways these two datasets were annotated. Most HELFI alignments are one-to-one, while many Blinker alignments are many-to-many: phrase-level alignments where every word in a source phrase is aligned with every word in a target phrase. This suggests that one can choose between GNN (TGDFA) and GNN (TGDFA+orig) based on the desired characteristics of the alignment.

5.1.1 Effect of Training Set Size

To investigate the effect of training set size, we train the GNN on subsets of our training data with increasing sizes. Figure 3 shows results. Performance improves fast until around 2,000 verses; then it stays mostly constant. Using more than 6,400 samples does not change the performance at all. Therefore, in the other experiments we use 6,400 randomly sampled verses from the training set to train GNNs.

[5] Note that we do not use any gold alignments for training the GNN. Using the verses from HELFI train split as our training set is for convenience. Our ablation experiment (Figure 3) show that a smaller subset of the training set is sufficient to achieve good performance

[6] pytorch-geometric.readthedocs.io

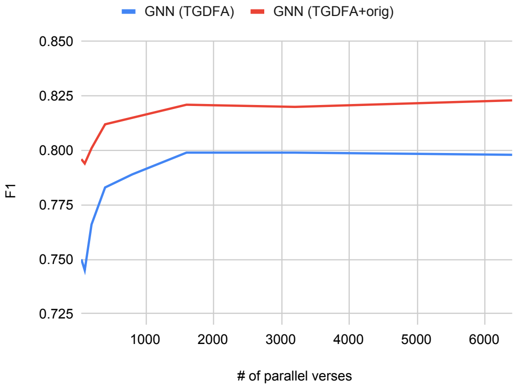


Figure 3: F_1 of GNN (TGDFA) and GNN (TGDFA+orig) on Blinker as a function of train size

5.1.2 Ablation Experiments

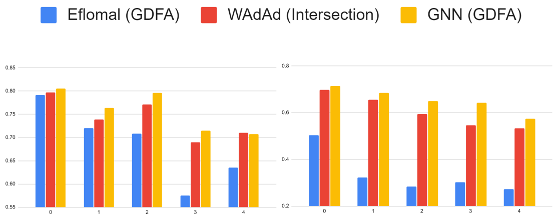
To examine the importance of node features, we ablate language, position, centrality, community and word embedding features. Table 3 shows that removal of graph structural features drastically reduces performance. Community features and language information are also important. Removal of word position information and word embeddings – which store semantic information about words – has the least effect. Based on these results, it can be argued that the lexical information contained in the initial alignments and in the community features provides a strong signal regarding word relatedness. The novel information that is crucial is about the overall graph structure which goes beyond the local word associations that are captured by word position and word embeddings.

5.1.3 Effect of Word Frequency

We investigate the effect of word frequency on alignment performance where frequency is calculated based on the source word in the PBC; the first bin has the highest frequency. Figure 4 shows that the performance of Eflomal drops with frequency and it struggles to align very rare words. In contrast, GNN is not affected by word frequency as severely and its performance gains are even greater for rare words. WAdad which is the multilingual baseline from (Imani Googhari et al., 2021) has the same trend as the GNN method, but the GNN method is more robust.

5.2 Annotation Projection

Table 4 presents accuracies for POS tagging in Yoruba. Unsupervised baseline performance is 50.86%. Supervised training using pseudo-labels mostly outperforms the unsupervised baseline. Projecting the majority POS labels to Yoruba improves over projecting English labels. Using the GNN model to project labels works best and outperforms Eflomal-GDFA-majority (resp. the unsupervised baseline) by 5% (resp. 15%) absolute improvement.



(a) ENG-FRA

(b) FIN-HEB

Figure 4: F_1 for different frequency bins.

	FIN-HEB	FIN-GRC	ENG-FRA
GNN (TGDFA)	0.720	0.780	0.804
¬ language	-0.323	-0.280	-0.370
¬ position	-0.068	-0.045	-0.066
¬ centrality	-0.636	-0.730	-0.772
¬ community	-0.204	-0.238	-0.253
¬ word-embedding	-0.139	-0.103	-0.129
GNN (TGDFA+orig)	0.651	0.758	0.824
¬ language	-0.238	-0.077	-0.162
¬ position	-0.088	+0.029	-0.032
¬ centrality	-0.556	-0.530	-0.617
¬ community	-0.156	-0.039	-0.083
¬ word-embedding	-0.135	+0.002	-0.058

Table 3: F_1 for GNNs and ΔF_1 for five ablations

6 Related Work

Bilingual Word Aligners. Much work on bilingual word alignment is based on probabilistic models, mostly implementing variants of the IBM models of Brown et al. (1993): e.g., Giza++ (Och and Ney, 2003), fast-align (Dyer et al., 2013) and Eflomal (Östling and Tiedemann, 2016). More recent work, including SimAlign (Jalili Sabet et al., 2020) and SHIFT-ATT/SHIFT-AET (Chen et al., 2020), uses pretrained neural language and machine translation models. Although neural models achieve superior performance compared to statistical aligners, they can only be used for fewer than two hundred high-resource languages that are supported by multilingual language models like BERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020). This makes statistical models the only option for the majority of the world's languages.

Model	Yoruba YTB
Unsupervised (Kondratyuk and Straka, 2019)	50.86
Eflomal Inter – eng	43.45
Eflomal GDFA – eng	55.13
Eflomal Inter – majority	54.13
Eflomal GDFA – majority	60.27
GNN (TGDFA) – majority	**65.74**
GNN (TGDFA+orig) – majority	64.55

Table 4: POS tagging with annotation projection for Yoruba. Apart from "Unsupervised", all lines show a sequence tagger trained on pseudo-labels induced by word alignments. GNN-based pseudo-labels outperform prior work by 5% absolute.

Multiparallel Corpora. Prior applications of using multiparallel corpora include reliable translations from small datasets (Cohn and Lapata, 2007), and phrase-based machine translation (PBMT) (Kumar et al., 2007). Multiparallel corpora are also used for language comparison (Mayer and Cysouw, 2012), typological studies (Östling, 2015; Asgari and Schütze, 2017) and PBMT (Nakov and Ng, 2012; Bertoldi et al., 2008; Dyer et al., 2013). ImaniGooghari et al. (2021) provide a tool to browse a word-aligned multiparallel corpus, which can be used for the comparative study of languages and for error analysis in machine translation.

To the best of our knowledge Lardilleux and Lepage (2008) and Östling (2014)[7] are the only word alignment methods designed for multiparallel corpora. However, the latter method is outperformed by Eflomal (Östling and Tiedemann, 2016), a bilingual method from the same author. Recently, Imani Googhari et al. (2021) proposed MPWA, which we use as our baseline.

Graph Neural Networks (GNNs) have been used to address many problems that are inherently graph-like such as traffic networks, social networks, and physical and biological systems (Liu and Zhou, 2020). GNNs achieve impressive performance in many domains, including social networks (Wu et al., 2020) and natural science (Sanchez-Gonzalez et al., 2018) as well as NLP tasks like sentence classification (Huang et al., 2020), question generation (Pan et al., 2020), summarization (Fernandes et al., 2019) and derivational morphology (Hofmann et al., 2020).

7 Conclusion and Future Work

We introduced graph neural networks and community detection algorithms for multiparallel word alignment. By incorporating signals from diverse sources as node features, including community features, our GNN model outperformed the baselines and prior work, establishing new state-of-the-art results on three PBC gold standard datasets. We also showed that our GNN model improves downstream task performance in low-resource languages through annotation projection.

We have only used node features to provide signals to GNNs. In the future, other signals can be added in the form of edge features to further boost the performance.

References

Željko Agić and Ivan Vulić. 2019. Jw300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 54–65, Berlin, Germany. Association for Computational Linguistics.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, pages 108–117, Copenhagen, Denmark. Association for Computational Linguistics.

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.

[7] github.com/robertostling/eflomal

Copenhagen, Denmark. Association for Computational Linguistics.

Necip Fazil Ayan and Bonnie J. Dorr. 2006. A maximum entropy approach to combining word alignments. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 96–103, New York City, USA. Association for Computational Linguistics.

Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, and Roldano Cattoni. 2008. Phrase-based statistical machine translation with pivot languages. In *International Workshop on Spoken Language Translation (IWSLT) 2008*.

Paolo Boldi and Sebastiano Vigna. 2014. Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262.

Ulrik Brandes. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177.

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.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735, Prague, Czech Republic. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Gennaro Cordasco and Luisa Gargano. 2010. Community detection via semi-synchronous label propagation algorithms. In *2010 IEEE international workshop on: business applications of social network analysis (BASNA)*, pages 1–8. IEEE.

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.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Patrick Fernandes, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Structured neural summarization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Victoria Fossum and Steven Abney. 2005. Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In *Second International Joint Conference on Natural Language Processing: Full Papers*.

Linton C Freeman. 1978. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239.

Michelle Girvan and Mark EJ Newman. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. *LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation*, page 639–648. Association for Computing Machinery, New York, NY, USA.

Valentin Hofmann, Hinrich Schütze, and Janet B. Pierrehumbert. 2020. A graph auto-encoder model of derivational morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1127–1138. Association for Computational Linguistics.

Lianzhe Huang, Xin Sun, Sujian Li, Linhao Zhang, and Houfeng Wang. 2020. Syntax-aware graph attention network for aspect-level sentiment classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 799–810, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matthias Huck, Diana Dutka, and Alexander Fraser. 2019. Cross-lingual annotation projection is effective for neural part-of-speech tagging. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 223–233, Ann Arbor, Michigan. Association for Computational Linguistics.

Ayyoob Imani Googhari, Masoud Jalili Sabet, Lutfi Kerem Senel, Philipp Dufter, François Yvon, and Hinrich Schütze. 2021. Graph algorithms for multiparallel word alignment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8457–8469, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ayyoob ImaniGooghari, Masoud Jalili Sabet, Philipp Dufter, Michael Cysou, and Hinrich Schütze. 2021. ParCourE: A parallel corpus explorer for a massively multilingual corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 63–72, Online. 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.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293.

Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. *International Workshop on Spoken Language Translation*.

Philipp Koehn, Franz J Och, and Daniel Marcu. 2003. Statistical phrase-based translation. Technical report, University of Southern California Marina del Rey Information Sciences Inst.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Shankar Kumar, Franz J. Och, and Wolfgang Macherey. 2007. Improving word alignment with bridge languages. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 42–50, Prague, Czech Republic. Association for Computational Linguistics.

Adrien Lardilleux and Yves Lepage. 2008. A truly multilingual, high coverage, accurate, yet simple, subsentential alignment method. In *The 8th conference of the Association for Machine Translation in the Americas (AMTA 2008)*, pages 125–132.

Omer Levy, Anders Søgaard, and Yoav Goldberg. 2017. A strong baseline for learning cross-lingual word embeddings from sentence alignments. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 765–774, Valencia, Spain. Association for Computational Linguistics.

William D. Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA. Association for Computational Linguistics.

Zhiyuan Liu and Jie Zhou. 2020. Introduction to graph neural networks. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(2):1–127.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Thomas Mayer and Michael Cysouw. 2012. Language comparison through sparse multilingual word alignment. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 54–62, Avignon, France. 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.

I. Dan Melamed. 1998. Manual annotation of translational equivalence: The Blinker project. *CoRR*, cmp-lg/9805005.

Preslav Nakov and Hwee Tou Ng. 2012. Improving statistical machine translation for a resource-poor language using related resource-rich languages. *Journal of Artificial Intelligence Research*, 44:179–222.

Mark EJ Newman. 2001. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132.

Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

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 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Robert Östling. 2014. Bayesian word alignment for massively parallel texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 123–127, Gothenburg, Sweden. Association for Computational Linguistics.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, Beijing, China. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1).

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1463–1475, Online. Association for Computational Linguistics.

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).

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. 2018. Graph networks as learnable physics engines for inference and control. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4470–4479. PMLR.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. 2014. Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1779–1785, Doha, Qatar. Association for Computational Linguistics.

Yongji Wu, Defu Lian, Yiheng Xu, Le Wu, and Enhong Chen. 2020. Graph convolutional networks with markov random field reasoning for social spammer detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1054–1061.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Anssi Yli-Jyrä, Josi Purhonen, Matti Liljeqvist, Arto Antturi, Pekka Nieminen, Kari M. Räntilä, and Valter Luoto. 2020. HELFI: a Hebrew-Greek-Finnish parallel Bible corpus with cross-lingual morpheme alignment. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4229–4236, Marseille, France. European Language Resources Association.

A Appendix

A.1 Languages

Afrikaans	Albanian	Arabic	Armenian	Azerbaijani	Bashkir
Basque	Belarusian	Bengali	Breton	Bulgarian	Burmese
Catalan	Cebuano	Chechen	Chinese	Chuvash	Croatian
Czech	Danish	Dutch	English	Estonian	Finnish
French	Georgian	German	Greek	Gujarati	Haitian
Hebrew	Hindi	Hungarian	Icelandic	Indonesian	Irish
Italian	Japanese	Javanese	Kannada	Kazakh	Kirghiz
Korean	Latin	Latvian	Lithuanian	Low Saxon	Macedonian
Malagasy	Malay	Malayalam	Marathi	Minangkabau	Nepali
Norwegian (B.)	Norwegian (N.)	Punjabi	Persian	Polish	Portuguese
Punjabi	Romanian	Russian	Serbian	Slovak	Slovenian
Spanish	Swahili	Sundanese	Swedish	Tagalog	Tajik
Tamil	Tatar	Telugu	Turkish	Ukrainian	Urdu
Uzbek	Vietnamese	Waray-Waray	Welsh	West Frisian	Yoruba

Table 5: List of the 84 languages we used in our experiments.

Chapter 5

Learning interpretable word embeddings via bidirectional alignment of dimensions with semantic concepts

Information Processing and Management 59 (2022) 102925

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Learning interpretable word embeddings via bidirectional alignment of dimensions with semantic concepts

Lütfi Kerem Şenel [a,*], Furkan Şahinuç [b,c,d], Veysel Yücesoy [c], Hinrich Schütze [a], Tolga Çukur [b,d], Aykut Koç [b,d]

[a] *Center for Information and Language Processing (CIS), LMU Munich, Germany*
[b] *Electrical and Electronics Engineering Department, Bilkent University, Ankara, Turkey*
[c] *ASELSAN Research Center, Ankara, Turkey*
[d] *National Magnetic Resonance Research Center (UMRAM), Bilkent University, Ankara, Turkey*

ARTICLE INFO

Keywords:
Word embeddings
Interpretability
Word semantics

ABSTRACT

We propose *bidirectional imparting* or *BiImp*, a generalized method for aligning embedding dimensions with concepts during the embedding learning phase. While preserving the semantic structure of the embedding space, BiImp makes dimensions interpretable, which has a critical role in deciphering the black-box behavior of word embeddings. BiImp separately utilizes both directions of a vector space dimension: each direction can be assigned to a different concept. This increases the number of concepts that can be represented in the embedding space. Our experimental results demonstrate the interpretability of BiImp embeddings without making compromises on the semantic task performance. We also use BiImp to reduce gender bias in word embeddings by encoding gender-opposite concepts (e.g., male–female) in a single embedding dimension. These results highlight the potential of BiImp in reducing biases and stereotypes present in word embeddings. Furthermore, task or domain-specific interpretable word embeddings can be obtained by adjusting the corresponding word groups in embedding dimensions according to task or domain. As a result, BiImp offers wide liberty in studying word embeddings without any further effort.

1. Introduction

Developments in machine learning lead to interdisciplinary studies and merge different research areas. An example can be observed in the natural language processing (NLP) based information science studies. There are increasingly improving information science studies that utilize NLP methods, especially word embeddings, while focusing on processing textual information. The scope of NLP-based studies can range from event detection (Qian et al., 2019; Tuke et al., 2020) to document retrieval (Bagheri et al., 2018). Computational studies on social media also frequently utilize NLP tools in various topics such as author profiling (López-Santillan et al., 2020), content processing (Moudjari et al., 2021; Roy et al., 2021) and hate speech detection (Pamungkas et al., 2021; Pronoza et al., 2021). What is common among these studies is that they all heavily depend on textual data. In representing and processing text, word embeddings play a key role and are used ubiquitously. Word embeddings are pre-trained semantic representations of words that hold numerous semantic features of natural languages. However, one disadvantage of word embeddings is that they learn language features as black-box schemes, unlike methods directly extracting determined and desired features. Therefore, studies on

[*] Corresponding author.
E-mail address: lksenel@cis.lmu.de (L.K. Şenel).

https://doi.org/10.1016/j.ipm.2022.102925
Received 26 October 2021; Received in revised form 6 February 2022; Accepted 1 March 2022
0306-4573/© 2022 Elsevier Ltd. All rights reserved.

their interpretability are of importance (Chen et al., 2016; Levy & Goldberg, 2014) for developing explainable NLP methods to be used in higher level information science applications.

Word embeddings (Bojanowski et al., 2017; Mikolov, Corrado et al., 2013; Mikolov, Sutskever et al., 2013; Pennington et al., 2014) – continuous dense vector representations – capture semantic and syntactic features of words. These embeddings are shown to be useful in a broad range of NLP applications involving topic modeling (Zhao et al., 2021), text classification (Elnagar et al., 2020), key-phrase extraction (Papagiannopoulou & Tsoumakas, 2018), document retrieval (Bagheri et al., 2018), named entity recognition (NER) (Nozza et al., 2021), query performance prediction (Roy et al., 2019), and extracting semantic features of words (Şahinuç & Koç, 2021). Although contextualized word embeddings and transformer-based architectures (Devlin et al., 2019; Radford et al., 2019; Vaswani et al., 2017) are becoming more and more prevalent due to their impressive performance on many NLP tasks, these models still use a static word embedding layer to represent input. Therefore, improvements to static word embeddings can potentially be transferred to contextual models as well (Schick & Schütze, 2020).

In addition to the traditional NLP tasks, word embeddings are frequently used in many other interdisciplinary domains. In neuroscience, they are employed to analyze the representation of semantics in brain activity (Huth et al., 2016; Ruan et al., 2016; Zhang et al., 2020) and as part of a decoder that extracts linguistic meaning from measured brain activity (Pereira et al., 2018). In psychiatry, they are used to detect incoherent speech for diagnosing schizophrenia (Iter et al., 2018; Voppel et al., 2021). In legal domain, they are used to predict outcomes of courts (Mumcuoğlu et al., 2021), evidence extraction from court records (Ji, Tao et al., 2020) and coreference resolution in legal texts (Ji, Gao et al., 2020). In the social domain, based on word, sentence and document embeddings polarization in social media can be analyzed (Demszky et al., 2019) and users of social media can be profiled (López-Santillan et al., 2020). Evolutionary linguists track historical changes in word meaning with embeddings (Hamilton et al., 2016; Kutuzov et al., 2018; Yüksel et al., 2021). Recent studies suggest that embeddings capture and quantify gender and ethnic biases in language (Bolukbasi et al., 2016; Caliskan et al., 2017; Garg et al., 2018) and their evolution over time (Agarwal et al., 2019).

Despite a large body of work on improved word embeddings (Bollegala et al., 2016; Celikyilmaz et al., 2015; Liu et al., 2015; Mrkšić et al., 2016; Yang & Mao, 2016; Yu & Dredze, 2014; Yu et al., 2017), a central limitation is their lack of interpretability: dimensions of the dense vector space do not individually represent semantic concepts (Chen et al., 2016; Levy & Goldberg, 2014) or other directly interpretable distinctions. Yet interpretability of word embeddings is highly desirable for several reasons. (i) It will enable researchers to make sense of embeddings of individual words, which are currently meaningful only in relation to other embeddings. (ii) Word embeddings serve as base representation in many deep learning models, so their interpretability is key for interpretable deep learning models. (iii) In interpretable embedding models, it is easier to remove redundant or nonrelevant dimensions, resulting in reduced computation and memory requirements. (iv) Interpretability also facilitates removal of gender, race and other biases (Dufter & Schütze, 2019).

Previous studies have put forth several important approaches to address limitations on interpretability of word embeddings. A group of studies proposed to use sparsity constraints such as non-negative matrix factorization (Fyshe et al., 2014; Luo et al., 2015; Murphy et al., 2012), sparse coding (Arora et al., 2018; Faruqui et al., 2015) and sparse auto-encoders (Subramanian et al., 2018) that yield sparse word representations. Since each word is represented by only a few dimensions, it is easier to understand what semantic features the dimensions capture. However, larger vocabulary requires higher dimensionality to achieve a desired sparsity level which increases memory and computation requirements. In addition, evaluations on common benchmark tests suggest that the resulting sparse embeddings often perform poorly compared to the dense embeddings that have distributed word representations. Another group of studies proposed to instead use orthogonal transformations over the high performing dense embeddings (Dufter & Schütze, 2019; Park et al., 2017; Zobnin, 2017) in order to preserve task performance. Yet, the level of improvement in interpretability that orthogonal transformations can achieve is relatively limited. Recently, in Şenel et al. (2020), we proposed an offline imparting approach to obtain interpretable word embeddings by modifying the objective function of GloVe (Pennington et al., 2014) to align each dimension of the vector space with a single pre-defined concept. However, this unidirectional imparting method does not utilize the full capacity of the embedding space (negative directions are ignored) and is limited to the training setting of the GloVe.

In this paper, we introduce *BiImp* (read as "bimp"), a generalized imparting approach that is capable of bidirectional imparting and online learning, hence more efficient and adaptable to new training data. BiImp utilizes both directions along each dimension of the vector space separately to encode two different concepts. The two concepts can be chosen arbitrarily or chosen as opposites (e.g., *good* – *bad*, *male* – *female*) as a special case (see Fig. 1), providing a more efficient use of the embedding space while increasing encoding flexibility. We demonstrate BiImp by modifying the word2vec skip-gram model (Mikolov, Corrado et al., 2013; Mikolov, Sutskever et al., 2013); concepts are selected from Roget's Thesaurus and WordNet. A hyperparameter can be tuned to achieve a good tradeoff between interpretability on the one hand and preservation of semantic structure on the other. We perform comprehensive experiments and demonstrate that interpretability of word embeddings improves while performance stays about the same. Inspired by Bolukbasi et al. (2016), we also demonstrate that BiImp can concentrate gender information in a single embedding dimension, the gender dimension, as a continuum. This supports efficient capture of gender bias and debiasing through removal of the gender dimension. In short, main outcomes of this study can be summarized as: (i) BiImp provides interpretable word embeddings by using both positive and negative directions of word embeddings; (ii) BiImp is compatible to different word embedding learning types; (iii) BiImp can be utilized to remove human biases from embeddings without compromising task performance.

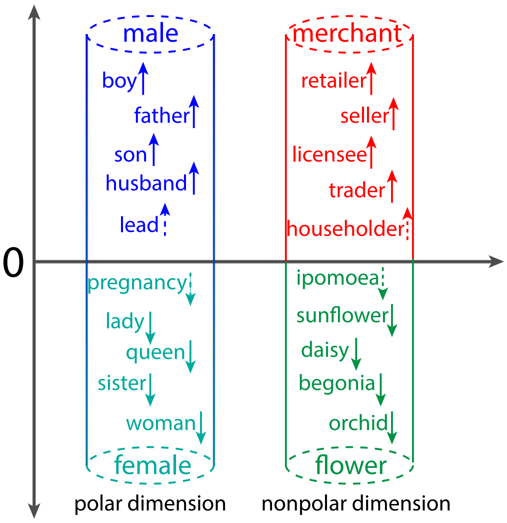


Fig. 1. Illustration of bidirectional imparting, the main idea underlying BiImp. The method increases interpretability of word embeddings by linking embedding dimensions to concepts. The concepts are taken from a conceptual resource that provides concepts along with word sets that are associated with them. BiImp "imparts" two concepts to each embedding dimension, one for the positive, one for the negative direction. E.g., the concepts "male" and "female" are associated with positive and negative directions of the polar dimension in the figure. Imparting is achieved by modifying the embedding training objective: during training, words associated with a concept are constrained to have high (or low) values on the dimension linked to the concept. As a result, the embedding vector of a word is directly interpretable: the value of each coordinate can be seen as a weight that the associated concept (positively or negatively associated concept) has in the representation of the meaning of the word. We study both polar dimensions (positive/negative concepts are opposites) and nonpolar dimensions (positive/negative concepts are unrelated). Solid arrow: word from resource. Dashed arrow: word not from the resource inferred to be related to the concept. We show that BiImp increases interpretability without impacting task performance and that it supports more effective debiasing.

2. Related work

2.1. Interpretability of word embeddings

Benefits of interpretable word embeddings have motivated several previous efforts to improve interpretability. Most of these studies introduce a sparsity constraint to learn sparse representations where each word is represented by only a few non-zero dimensions. The motivation behind sparsity is that by investigating the words that correspond to non-zero values in a dimension, one can infer which semantic features are encoded in that dimension. Based on this idea, Murphy et al. (2012) propose non-negative sparse embeddings (NNSE) to perform non-negative matrix factorization (NMF) on word co-occurrence variant matrices. As an extension to NNSE, Fyshe et al. (2014) proposed joint non-negative sparse embeddings (JNNSE) to incorporate additional knowledge on word similarity as measured by the similarity of cortical activity patterns. To address the memory and scale issues of NNSE-based methods, Luo et al. (2015) proposed an online learning method, where sparse embeddings were obtained using a modified skip-gram model (Mikolov, Sutskever et al., 2013). Several other studies proposed to learn sparse transformations that map pretrained state-of-the-art embeddings to sparse, more interpretable vector spaces instead of learning them from corpora (or co-occurrence matrices) directly. Arora et al. (2018) and Faruqui et al. (2015) use sparse coding methods and Subramanian et al. (2018) train a sparse auto-encoder. Inspired by research in topic modeling, Panigrahi et al. (2019) proposed a method named Word2Sense based on the Latent Dirichlet Allocation (LDA) to extract distributions of difference word senses from a corpus, which are then used to learn sparse interpretable word embeddings. While the above-mentioned approaches can increase interpretability to a certain degree, they do not exercise control over the specific concepts or word senses that are captured in the embedding dimensions.

Sparse representations typically have higher dimensionality than dense embeddings since only a few words are encoded in each dimension. Thus, they can suffer from memory and scaling issues especially for tasks that require a large vocabulary. To strictly preserve the dimensionality and semantic structure of word embeddings, several researchers proposed orthogonal instead of sparse transformations. Park et al. (2017) experimented with rotation algorithms based on exploratory factor analysis (EFA) with orthogonality constraints. Zobnin (2017) used orthogonal transformations to improve clustering of words along individual embedding dimensions. However, increases in clustering along a subset of embedding dimensions come at the expense of reduced clustering (i.e., interpretability) along the remaining dimensions (Zobnin, 2017). Dufter and Schütze (2019) and Rothe and Schütze (2016) use orthogonal transformations to align a linguistic signal (e.g., a collection of words) to an embedding dimension to obtain an interpretable subspace. However, this method has only been demonstrated in a low-dimensional subspace to date, so its performance in higher dimensional subspaces remains unclear. In a concurrent, independent study (Mathew et al., 2020), the transformation method *POLAR* was proposed to map an existing embedding space to a polar space where each embedding dimension corresponds to a pair of antonyms (i.e., polar opposites). In a recent study (Şenel et al., 2020), an imparting method was proposed in which individual dimensions of the model were aligned with concepts defined a priori based on an external resource. Şenel et al. (2020) demonstrated the effectiveness of this method only for the offline GloVe method, and only the positive direction of each dimension was matched up with a concept.

2.2. Gender bias

Ensuring the fairness of mathematical models is one of the most crucial issues in machine learning based information processing. The roles of machine learning and artificial intelligence have an increasing momentum in many real-world applications such as job hiring, granting loans, college applications (Makhlouf et al., 2021). Therefore, algorithms, model parameters, or model features must not include gender, race, ethnic or any other unwanted bias. In Makhlouf et al. (2021), important notions of fairness related to real-world scenarios are extracted, and necessary fairness notions are recommended for each specific setup that includes machine learning.

Bolukbasi et al. (2016) is one of the pioneering studies that investigate gender bias in word embeddings. Authors realize that some occupations that are supposed to be gender-neutral are mapped in favor of one gender by word embeddings. For example, word `man` is closer to `programmer` than `woman` in semantic space. To eliminate this problem, the authors propose two different debiasing methods named soft debiasing and hard debiasing, respectively (Bolukbasi et al., 2016). Caliskan et al. (2017) show that training datasets can unintentionally involve not only gender bias but also morally neutral biases. They also propose the Word Embedding Association Test (WEAT) and the Word Embedding Factual Association Test (WEFAT) to quantify the bias present in texts. In the diachronic study of Garg et al. (2018), it is shown that word embeddings that are trained on different texts from different timelines can reflect social, demographic, and cultural features of the corresponding period. On the other hand, Gonen and Goldberg (2019) approach this issue in a critical way by claiming that the proposed debiasing methods in the literature are not sufficient to remove the bias completely, and that the debiasing methods provide superficial cleaning, and this problem should be dealt with in-depth. The recent advances come with the requirement of detecting and removing biases in contextualized word embeddings and language models. To this end, Liang et al. (2020) propose SENT-DEBIAS method that reduces the social biases in sentence level representations. The proposed method is performed in BERT and ELMO models as an extension of hard debasing in Bolukbasi et al. (2016).

Gender bias is not limited to exist only in word embeddings. Recommender systems and search engines also host gender bias in various ways. Melchiorre et al. (2021) investigate the gender fairness in recommendation algorithms in the music domain. The authors demonstrate the gender inequality in the recommendation performance in favor of the male user group. In addition, they also show that applying debiasing algorithms are beneficial for the improvement of gender fairness. On the other hand, Fabris et al. (2020) propose a measure named 'Gender Stereotype Reinforcement' to evaluate the tendency of search engines to support gender stereotypes. The effect of the embedding debiasing methods on search engines is also inspected.

Detecting gender discrimination is also as important as eliminating gender bias. There exist many kinds of hate speech in social media (Kocoń et al., 2021). Identifying such expressions that contain hatred and biased patterns is also a significant subject of information processing. For instance, Pamungkas et al. (2020) present a review of the state-of-the-art misogyny detection. The most predictive language features for distinguishing hatred and biased content are also presented. Learning these features takes an important part in both detecting and eliminating gender bias in machine learning-based information processing models.

3. Research objectives

Our main contributions and research objectives are as follows:

- We propose BiImp, a bidirectional imparting algorithm to improve interpretability of word embeddings that utilizes both directions of each embedding dimension separately to encode different concepts.
- We demonstrate that the bidirectional imparting of arbitrary concepts offers superior performance compared to encoding of polar opposites to each embedding dimension, in terms of interpretability, intrinsic and downstream evaluation tasks.
- We perform comprehensive evaluations and provide comparison with previous work, showing that BiImp achieves greater interpretability without sacrificing performance.
- We propose for the first time an imparting method to concentrate gender information to a designated embedding dimension, along with an hybrid method that achieves concurrent gender and interpretability imparting. We show that this dimension effectively captures gender information and improves the performance of gender debiasing methods, in terms of gender bias metrics and high-level evaluation tasks.

4. Methods

4.1. Imparting

Unidirectional imparting (*UniImp*) is a method that enhances interpretability in GloVe word embeddings by forcing words related to predefined concepts to project more strongly onto individual embedding dimensions (Şenel et al., 2020). To achieve this, GloVe's cost is modified as follows:

$$\begin{aligned}&\sum_{i,j=1}^{V} f(X_{ij}) \Bigg[\left(\vec{w}_i^T \vec{\tilde{w}}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2 \\ &+\ k^g \left(\sum_{c=1}^{C} \mathbb{1}_{i\in F_c}\ g(w_{i,c}) + \sum_{c=1}^{C} \mathbb{1}_{j\in F_c}\ g(\tilde{w}_{j,c}) \right) \Bigg]\end{aligned} \tag{1}$$

where $\vec{w}_i$ and $\vec{\tilde{w}}_j$ denote word and context vectors, $w_{i,c}$ and $\tilde{w}_{j,c}$ denote the cth components of word and context vectors, b_i and $\tilde{b}_j$ denote word and context biases, X_{ij} denotes co-occurrence of the ith and jth words in the vocabulary, V denotes vocabulary size, and $f(\cdot)$ is a weighting function to prevent bias from rare words. The first term in the cost is GloVe's original cost function. It aims to capture semantic structure in the embedding model based on word co-occurrences. The second term aims to align embedding dimensions with word-groups. In this latter term, C denotes the number of word-groups ($C \le dim(\vec{w})$), $\mathbb{1}_{x\in S}$ is the indicator variable for the inclusion $x \in S$, F_c denotes the indices of words that belong to the cth group, k_g controls the relative weighting of the second term, and $g(\cdot)$ is a monotone decreasing function that adjusts the size of the updates during training. $g(\cdot)$ is defined as:

$$g(x) = \begin{cases} 1/2 \cdot exp(-2x), & \text{if } x < 0.5 \\ 1/(4ex), & \text{otherwise.} \end{cases}$$

4.2. Generalized bidirectional imparting

In this paper, we propose BiImp, a generalized imparting framework that is capable of online learning and bidirectional imparting. To alleviate computation and memory limitations, we focus on the skip-gram model of word2vec with negative sampling. The objective that the skip-gram model aims to maximize for a word pair (i, j) is given as:

$$\log\ \sigma(\vec{\tilde{w}}_j{}^T \vec{w}_i) + \sum_{t=1}^{m} \mathbb{E}_{z_t\sim P_n(w)} \left[log\ \sigma(-\vec{\tilde{w}}_{z_t}{}^T \vec{w}_i) \right]. \tag{2}$$

Although the learning mechanisms of GloVe and word2vec are different, unidirectional imparting can still be implemented by maximizing the following modified objective:

$$\begin{aligned}&\log\ \sigma(\vec{\tilde{w}}_j{}^T \vec{w}_i) + \sum_{t=1}^{m} \mathbb{E}_{z_t\sim P_n(w)} \left[\log\ \sigma(-\vec{\tilde{w}}_{z_t}{}^T \vec{w}_i) \right] \\ &\quad -\ k^w \Big(\sum_{c=1}^{C} \mathbb{1}_{i\in F_c}\ g(w_{i,c}) + \sum_{c=1}^{C} \mathbb{1}_{j\in F_c}\ g(\tilde{w}_{j,c}) \Big).\end{aligned} \tag{3}$$

In objectives (2) and (3), σ is the sigmoid function, m is number of negative samples and $P_n(w)$ is the unigram distribution ($U(w)$) raised to the power 3/4, and z_t is the index of the word from the tth draw from the unigram word distribution. Although the additional terms in (1) and (3) look identical, throughout the training process, their relative influence over the original embedding loss can be significantly different. To account for these differences, different weighting factors k^g and k^w are defined.

Imparting was previously only performed for the positive direction of embedding dimensions. But negative directions are equally suitable to encode semantic, interpretable concepts. Based on this argument, we extend the imparting method to both directions of the embedding dimensions. Given a fixed number for embedding dimensions, BiImp doubles the concept capacity compared to the unidirectional case. Moreover, by aligning opposite concepts such as *good* and *bad* or *male* and *female* with opposing directions of the same dimension, these concepts can be represented in a continuum.

The proposed objective for BiImp, the bidirectionally imparted word2vec model is as follows:

$$\begin{aligned}&\log\ \sigma(\vec{\tilde{w}}_j{}^T \vec{w}_i) + \sum_{t=1}^{m} \mathbb{E}_{z_t\sim P_n(w)} \left[\log\ \sigma(-\vec{\tilde{w}}_{z_t}{}^T \vec{w}_i) \right] \\ &-\ k^w \ \Bigg(\sum_{c=1}^{C^+} \mathbb{1}_{i\in F_c^+}\ g(w_{i,c}) + \sum_{c=1}^{C^+} \mathbb{1}_{j\in F_c^+}\ g(\tilde{w}_{j,c}) \\ &\quad - \sum_{c=1}^{C^-} \mathbb{1}_{i\in F_c^-}\ g(w_{i,c}) - \sum_{c=1}^{C^-} \mathbb{1}_{j\in F_c^-}\ g(\tilde{w}_{j,c}) \ \Bigg)\end{aligned} \tag{4}$$

where C^+ and C^- are the number of word-groups associated with positive and negative directions respectively ($C^+ \le dim(\vec{w})$, $C^- \le dim(\vec{w})$). F_c^+ and F_c^- denote the indices of words that belong to the cth group in the positive and negative directions, respectively.

Here word-groups encoded in opposing directions of a given dimension are referred to as word–group pairs. Ideally, the word–group pairs should not contain overlapping words ($F_c^+ \cap F_c^- = \emptyset\ \ \forall c$) to prevent weak word representations. In practice, this problem can be alleviated by rearrangement of word–group pairs. In this study, we apply the following simple rearrangement procedure to prevent overlap. For a given embedding dimension, we first select two random word-groups. When overlap is present, the second word–group is reselected from the set of remaining unpaired word-groups. This procedure is iterated until all word-groups are paired.[1]

4.3. Lexical resources

The imparting method requires an external lexical resource that constitutes a basis for interpretability. A trivial interpretation of an embedding model is possible if each embedding dimension is aligned with a distinct concept, (i.e., a word-group). Since practical embedding models can have variable dimensionality, a broad lexical resource that can be used to flexibly extract an arbitrary number of concepts is desirable. To this end, we utilized two lexical resources which are the Roget's Thesaurus (Roget, 2008) and the WordNet (Miller, 1995).

In Şenel et al. (2020), Roget's Thesaurus is utilized as an external resource. Roget's Thesaurus follows a tree structure, where the actual words and phrases are grouped under 1,000 categories making the leaves of the tree structure. We extract word-groups from the thesaurus by partitioning the tree structure starting at the root node from which all other nodes descend. A threshold λ^r_{max} is set for the maximum size of a node. Size of a given node is defined as the number of unique descendant words. During partitioning, each node with size less than the threshold is selected to define a word–group, which consists of descendant words for that node. For an above-threshold node without any children nodes, the word–group was defined as the λ^r_{max} descendant words with the highest-frequency ranks. Among the resulting word-groups, the ones that contain less than λ^r_{min} words are discarded. Finally, word-groups are constructed after discarding the groups with the largest median frequency ranks (i.e., groups that contain more rare words on average).

In addition to the Roget's Thesaurus, we investigate another important lexical resource that can be used to extract semantic word-groups, the WordNet (Miller, 1995). WordNet is a popular lexical database for English in which nouns, verbs, adjectives and adverbs are grouped together into synsets. Each synset expresses a distinct concept. Synsets are interlinked based on their semantic and lexical relations creating a network of related words and concepts. WordNet is similar to a thesaurus since it can be used to group words together based on meaning. However, there are two important differences between WordNet and a thesaurus. First, the network in WordNet is not based on word forms (i.e., sequence of letters) but on specific senses of words. As such, different senses of a word are represented by different synsets providing semantic disambiguation. Second, semantic relations between words are labeled in WordNet to describe the relation types, unlike a thesaurus where words are grouped merely based on similarity in meaning. WordNet is a comprehensive lexical resource containing 117,000 synsets each of which is linked to other synsets. The most frequently encoded relation between synsets is the super-subordinate relation (also known as hyper-hyponymy) that links more general synsets like *furniture* to increasingly specific synsets like *bed* and *bunkbed*. In other words, the category *furniture* includes *bed*, the category *bed* includes *bunkbed* and so on. In the hierarchical structure of WordNet, all noun synsets ultimately go up the root node *entity*.

4.4. Interpretability evaluation

Following Şenel et al. (2020), we evaluate the interpretability of the word embeddings based on SEMCAT categories (Şenel, Utlu et al., 2018) and subcategories (Şenel, Yücesoy et al., 2018). SEMCAT (sub)categories are taken as an approximation for the semantic concepts that humans can use to interpret embedding dimensions. Based on SEMCAT, we calculate the *Interpretability Score IS*, which is a measure of how strongly these (sub)categories are represented in embedding dimensions. This metric is low-cost, fast, reproducible and was shown to correlate well with human judgement (Şenel et al., 2020). However, it cannot capture the difference between interpretability changes in the positive and negative directions of an embedding dimension because it performs maximum pooling over the opposite directions of each dimension. To capture this information, we propose a new directional interpretability score:

$$\begin{aligned}
IS^+_{l,k} &= \max_{n_{min}\le n\le n_k} \frac{|S_k \cap V_l^+(\lambda\times n)|}{n} \times 100\\
IS^-_{l,k} &= \max_{n_{min}\le n\le n_k} \frac{|S_k \cap V_l^-(\lambda\times n)|}{n} \times 100\\
IS^+_l &= \max_k IS^+_{l,k}, \quad IS^-_l = \max_k IS^-_{l,k},\\
IS^+ &= \frac{1}{D}\sum_{l=1}^{D} IS^+_l, \quad IS^- = \frac{1}{D}\sum_{l=1}^{D} IS^-_l
\end{aligned} \tag{5}$$

In Eq. (5), $IS^+_{l,k}$ and $IS^-_{l,k}$ represent the interpretability scores in the positive and negative directions of the lth dimension ($l \in \{1,2,\dots,D\}$, $D = dim(\vec{w})$) for the kth category ($k \in \{1,2,\dots,K\}$, $K = 110$) in SEMCAT, respectively. S_k is the set of words in the kth category in SEMCAT and n_k is the number of words in S_k. n_{min} is the minimum number of words required to construct a semantic category (i.e., to represent a concept). $V_l(\lambda \times n)$ represents the set of $\lambda \times n$ words that have the highest (V_l^+) and lowest (V_l^-) values in the lth dimension of the embedding space. For all evaluations we use $\lambda = 5$.

[1] For cases when word-groups have a substantial proportion of overlapping words, more sophisticated matching algorithms might be necessary. However, here, we were able to find a non-overlapping pairing after a few trials (less than 5).

4.5. Gender bias

4.5.1. Intrinsic bias evaluation

BiImp matches each dimension with concepts and thereby makes it interpretable: it now clearly represents specific concepts. As Dufter and Schütze (2019) argue, this important property can facilitate removal of unwanted information from the model. A common example of such undesirable information is the inherent gender bias in corpora that is reflected in learned embedding models. Bolukbasi et al. (2016) report that embedding models often contain gender bias, particularly for occupation related words.

As discussed in Section 4.2, an important advantage of BiImp over unidirectional imparting is that two concepts with opposite meanings can be represented in a single dimension as a continuum. Since the concepts *male* and *female* are opposites, they can be encoded in the opposite directions of the same dimension, creating a continuous gender dimension. The gender components of words can then be inferred directly from their projections onto the gender dimension. To create a gender dimension, we construct two word-groups corresponding to *male* and *female* concepts using (Bolukbasi et al., 2016)'s gender-specific word set S of 291 professions.

Bolukbasi et al. (2016) proposed two different measures to assess level of gender bias in word embeddings, namely direct bias and indirect bias. Here, we use the direct bias measure:

$$b_{\kappa}^{direct} = \frac{1}{|N|} \sum_{w \in N} |\cos(\vec{w}, \vec{g})|^{\kappa} \tag{6}$$

where N is the set of gender neutral words, $\vec{g} = \vec{w}_{she} - \vec{w}_{he}$ is the gender vector and κ is a parameter that controls the relative weighting of high vs. low bias levels; we set $\kappa = 1$. Gender neutral words were obtained by taking the complement of Bolukbasi et al. (2016)'s gender-specific word set S such that $N = W \setminus S$ where W is the set of all words.

To evaluate BiImp on gender bias, we use the stereotypical gender bias levels b^s provided by Bolukbasi et al. (2016) for S (the set of 291 profession words), which were obtained by human assessment.[2] We calculate the correlation B^g between stereotypical biases b^s and the biases b^{direct} based on Eq. (6):

$$B^g = \text{corr}(b^s, b^{direct}) \tag{7}$$

as well as the correlation B^{gd} between the stereotypical biases b^s and the biases b^{gd} from the gender dimension:

$$B^{gd} = \text{corr}(b^s, b^{gd}) \tag{8}$$

where b^{gd} is calculated as:

$$b_p^{gd} = \begin{cases} \min\left(1, \frac{w_p}{\mu_m}\right) & \text{if } w_p \geq 0, \\ -\min\left(1, \frac{w_p}{\mu_f}\right) & \text{if } w_p < 0, \end{cases} \tag{9}$$

μ_m and μ_f are the average values of the words in the *male* and *female* word-groups in the gender dimension (gd), respectively. w_p stands for the value of the pth profession in the gender dimension. The intuition behind Eq. (9) is that we want a value between -1 and 1 (the range of b^s) to indicate level of bias. We could map the entire range of values on the dimension to the interval $[-1, 1]$, but that would give too much weight to outliers. We therefore use μ_m/μ_f as upper/lower bounds for w_p. B^{gd} (resp. B^g) indicate how well the BiImp gender dimension (resp. the gender vector $\vec{g}$) captures stereotypical gender bias.

4.5.2. Reducing gender bias

Bolukbasi et al. (2016) proposed two methods for gender debiasing: namely *hard debiasing (neutralize and equalize)*, and *soft bias correction*. Here, we consider the hard debiasing method, where the gender subspace is first identified via the principal component analysis (PCA). To do this, difference between word vectors of 10 pairs of gender words (i.e., *female–male, she–he, girl–boy*, etc.) were computed, and PCA was then performed on these 10 difference vectors. The principal component with the largest eigenvalue predominantly captures variance among the difference vectors (around 60% of total variance), suggesting that gender bias primarily lies along a single direction in the embedding space. In the *neutralize* stage, vectors for the gender-neutral words are updated to ensure that their projections onto the first principal component (i.e., gender subspace) is zero. Equality sets are then defined where each set contains a gender pair such as *{men, women}*. In the *equalize* stage, vectors of the words in the equality sets are updated such that the gender pair in each set becomes equidistant to the gender subspace. Therefore, following the equalization stage, each gender-neutral word becomes equidistant to both *men* and *women* vectors.

In this work, we investigate the effect of concentrating gender information in a single dimension of the embedding model via bidirectional imparting. We employ a two-stage approach for reducing gender bias in imparted embedding models. First, we remove the gender dimension from the embedding model to cancel out gender bias as suggested in Dufter and Schütze (2019). Since creation of a gender dimension concentrates gender information in a single dimension, removal of this dimension is expected to remove gender bias from the entire model. Next, we perform hard debiasing as described in Bolukbasi et al. (2016) on the reduced embedding model. Quantitative comparisons of bias level are performed on imparted and reduced embedding models both prior to and after debiasing procedures.

[2] https://github.com/tolga-b/debiaswe/blob/master/data/professions.json Professions that were not in our vocabulary were filtered out.

Table 1
Summary statistics of the word–group datasets.

Word counts	Roget's Thesaurus		WordNet	
	(300 grp.)	(600 grp.)	(300 grp.)	(600 grp.)
Total	20 978	40 350	26 964	18 965
Unique	12 289	19 870	18 123	13 853
Average	69.9 ± 53.7	67.3 ± 54.6	89.9 ± 74.2	31.6 ± 15.9

4.5.3. Bias in classification

Prost et al. (2019) argue that lower gender bias levels as measured by Eq. (6) do not always translate to reduced gender bias in classification. We therefore also evaluate on *BiosBias* (De-Arteaga et al., 2019), a classification dataset of 397,907 biographies extracted from CommonCrawl. Each biography is annotated as male or female and as being one of 28 different occupations. The task is to classify each subject's occupation given their biography. The train/dev/test split is 258,640/39,790/99,477.

For occupation classification based on an embedding model, single words in a given biography are first projected to the embedding space. Each biography is thereby represented as the average vector of words within the biography. A linear classifier with softmax output is used, and hyperparameters are tuned based on validation set performance. Classification accuracy is used as the performance measure. As a latent measure of gender bias in embedding models, fairness of the classifier to the two genders are examined as described in Hardt et al. (2016) as equality of opportunity. Specifically, we measure the True Positive Rate Gender Gap ($\mathrm{TPR_{gap}}$) and True Negative Rate Gender Gap ($\mathrm{TNR_{gap}}$) for the classifier. $\mathrm{TPR_{gap}}$ for a given occupation is measured as:

$$\begin{aligned}\mathrm{TPR}_{o,\mathrm{gap}} = |&Pr\{\hat{B}_o = 1|B_o = 1, B_g = f\} - \\ &Pr\{\hat{B}_o = 1|B_o = 1, B_g = m\}|,\end{aligned} \tag{10}$$

where o is an occupation, B_o ($\hat{B}_o$) is the (estimated) occupation of a biography and B_g its gender (m/f = male/female). $\mathrm{TPR_{gap}}$ (resp. $\mathrm{TNR_{gap}}$) is the difference in accuracy between the two genders of detecting the presence (resp. absence) of an occupation. We interpret this as a measure of the gender fairness of the word embeddings for o. We compute $\mathrm{TPR_{gap}}/\mathrm{TNR_{gap}}$ as the average over all $\mathrm{TPR}_{o,\mathrm{gap}}/\mathrm{TNR}_{o,\mathrm{gap}}$.

5. Experiments and results[3]

In this section, we describe our experiments and present our findings. Section 5.1 describes how we extract word groups from lexical resources. Section 5.2 describes our main experiments for improving interpretability and presents our findings. Section 5.3 presents our gender debiasing experiments. Section 5.4 evaluates the performance of gender de-biased embeddings, and Section 5.5 presents a hybrid gender and interpretability imparted model.

5.1. Word–group extraction

We investigate two lexical resources to extract word groups for imparting: Roget's Thesaurus (Roget, 2008) and WordNet (Miller, 1995). To extract word groups from Roget's Thesaurus, we follow the extraction procedure in Şenel et al. (2020) and extract 300 and 600 word groups by taking $\lambda^w_{min} = 20$ and $\lambda^w_{min} = 15$, respectively. To extract word-groups from WordNet, we follow a similar procedure and partition the hierarchical structure starting from the root node. We follow an iterative approach, where the largest node is divided to its hyponyms in each iteration. Node size is taken as the number of unique words descending from a node after filtering based on the vocabulary extracted from Wikipedia. We discard the nodes with size less than λ^w_{min}. Iterations are stopped when the number of nodes exceeds the desired word–group count. Note that the desired word-group count may not be achieved if λ^w_{min} is selected too large. The groups with the smallest number of member words are discarded to achieve desired word-group. We take $\lambda^w_{min} = 25$ and $\lambda^w_{min} = 15$ for 300 and 600 WordNet word groups, respectively. Table 1 summarizes the statistics for the constructed word-groups.

5.2. Interpretability enhancement

Our training corpus is the English Wikipedia. To pre-process the Wikipedia dump, all document numbers, URLs, HTML syntax and non-alphanumeric characters are cleared. Remaining words are lower-cased. Resulting corpus consists of 2,127,511,369 tokens. Words with less than 100 occurrences are discarded from the corpus. The final vocabulary contains 229,922 unique words (types). To test generalizability of imparting approach, using the 300 Roget word groups and the objectives Eq. (1) and (3), we train two sets of 300-dimensional unidirectionally imparted embeddings (one for GloVe one for word2vec) for different k^g and k^w values. We measure their interpretability using IS^+ (Eq. (5)). Fig. 2 shows interpretability scores for unidirectionally imparted GloVe and word2vec in the positive direction for $n_{min} = 5$ and $n_{min} = 10$. These results suggest that regularization term for imparting is viable for word2vec algorithm as well. However, original word2vec embeddings have lower interpretability values than original GloVe embeddings and word2vec requires stronger regularization than GloVe ($k^w > k^g$) to achieve similar interpretability.

[3] Data and codes are provided at: https://github.com/lksenel/biimp.

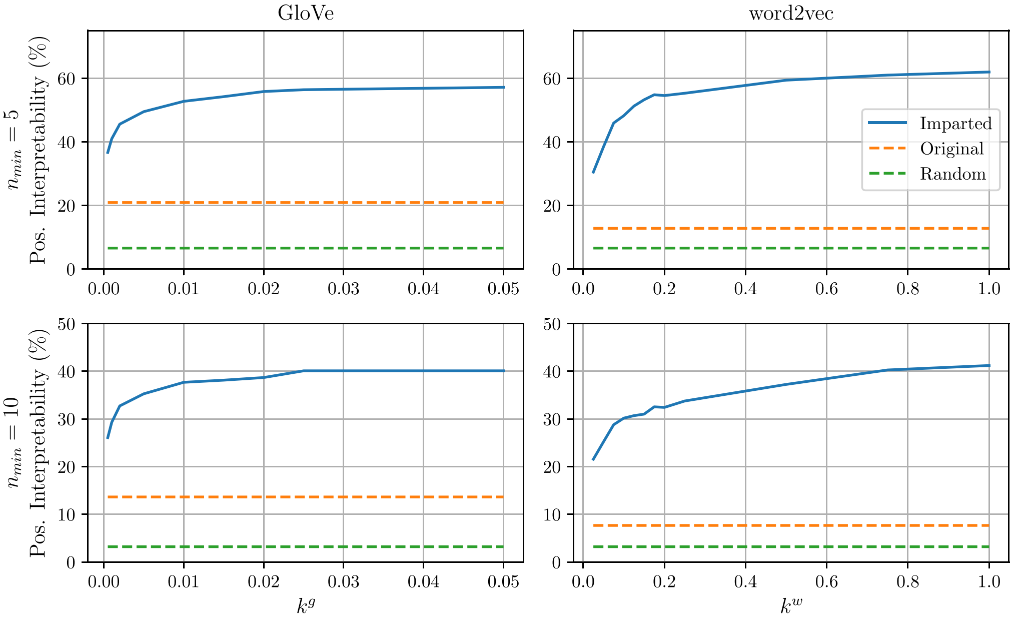


Fig. 2. Interpretability scores in the positive direction (IS^+) using $n_{min} = 5$ (top row) and $n_{min} = 10$ (bottom row) for unidirectionally imparted GloVe (left column) and word2vec (right column) algorithms for $k^g \in [0.0005, 0.05]$ and $k^w \in [0.025, 1.00]$, respectively. Interpretability scores for original embeddings and a random baseline are displayed for comparison as orange and green dashed lines, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Then, using the 600 word-groups from Roget's Thesaurus and WordNet, we investigate the viability of bidirectional imparting for Word2Vec. Using the objective Eq. (4), we train two sets of 300-dimensional BiImp vectors (one for Roget's and WordNet each) for different k^w values. We additionally train word2vec vectors without bidirectional imparting. For the training all imparted models and the original word2vec model, we use `VOCAB_MIN_COUNT` = 100, `MAX_ITER` = 15, `WINDOW_SIZE` = 8, `NEGATIVE` = 15, `SAMPLE` = 10^{-4}.

We evaluate the resulting embeddings on two measures: interpretability scores IS$^+$ and IS$^-$ (Eq. (5)) and intrinsic performance, based on word similarity[4] (Faruqui & Dyer, 2014) and word analogy[5] (Mikolov, Corrado et al., 2013) tests. Fig. 3 shows interpretability values of the unidirectionally and bidirectionally imparted word2vec embeddings using Roget and WordNet word-groups for $n_{min} = 5$ and $n_{min} = 10$ in both of the positive and negative directions. Bidirectional imparting achieves considerably improved interpretability compared to unidirectional imparting in the negative direction with minimal compromise in the positive direction.

Fig. 4 presents the performances of the embeddings on word similarity and word analogy tests. Performance decreases with increasing k^w. However, for bidirectional imparting of WordNet word-groups, performance is on par with original embeddings for $k^w \leq 0.2$. While WordNet word-groups somewhat reduce interpretability compared to Roget word-groups in bidirectional setting, they are much better at preserving the semantic structure of the embedding space as suggested by similarity and analogy tests. Taken together, results in Figs. 3 and 4 suggest that bidirectional imparting of WordNet word-groups at relatively low k_w is the optimal setting for word2vec. Therefore, we use WordNet-based BiImp in the rest of the paper.

5.2.1. Interpretability comparison

We compare BiImp with six state-of-the-art methods for interpretability enhancement: OIWE-IPG (Luo et al., 2015), SOV (Faruqui et al., 2015), Parsimax (Park et al., 2017), Word2Sense (Panigrahi et al., 2019) POLAR (Mathew et al., 2020) and UniImp (Şenel et al., 2020). We do not consider SPINE (Subramanian et al., 2018) because it scaled poorly for large vocabularies in our experiments.

OIWE-IPG was trained on the same corpus as the word2vec embeddings using the default parameters reported in Luo et al. (2015), yielding 300 dimensional vectors. SOV and Parsimax that work on pretrained embeddings were performed on the original word2vec embeddings, again using suggested parameters in Faruqui et al. (2015) and Park et al. (2017), resulting in 1000 and 300 dimensional vectors, respectively. For Word2Sense, we used the publicly available 2250 dimensional pretrained vectors[6] due to computational

[4] Word similarity results were averaged across 13 datasets: WS-353-ALL, SIMLEX-999, VERB-143, SimVerb-3500, WS-353-REL, RW-STANFORD, YP-130, MEN-TR-3k, RG-65, MTurk-771, WS-353-SIM, MC-30, MTurk-287.

[5] http://download.tensorflow.org/data/questions-words.txt.

[6] https://github.com/abhishekpanigrahi1996/Word2Sense.

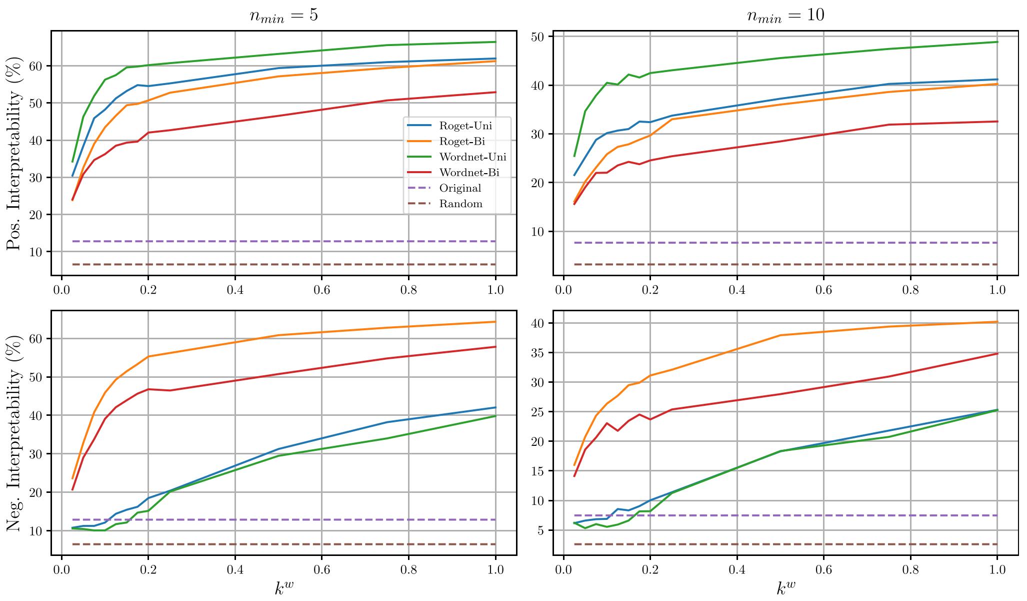


Fig. 3. Positive (top) and negative (bottom) direction interpretability scores for unidirectionally imparted word2vec embeddings using Roget's Thesaurus (Roget-Uni) and WordNet (WordNet-Uni) and their bidirectionally imparted versions (BiImp (Roget), BiImp (WordNet)) for $k^w \in [0.025, 1.00]$ along with the original word2vec embedding and a random baseline for $n_{min} = 5$ (left) and $n_{min} = 10$ (right).

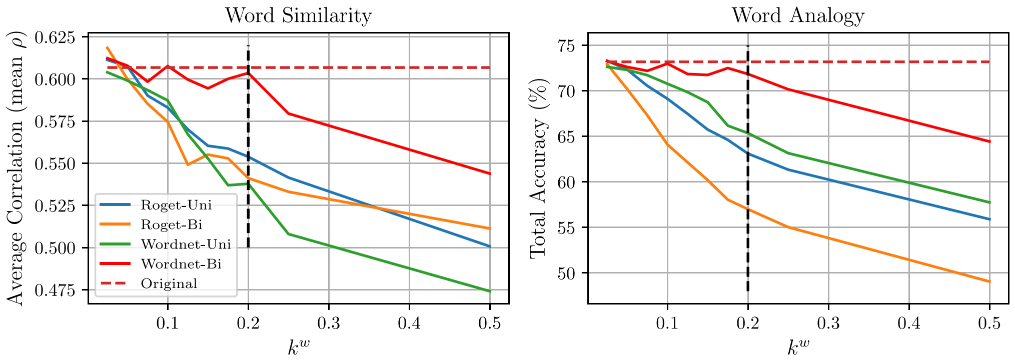


Fig. 4. Performance of unidirectionally imparted word2vec embeddings using Roget's Thesaurus (Roget-Uni) and WordNet (WordNet-Uni) and their bidirectionally imparted versions (Roget-Bi, WordNet-Bi) for $k^w \in [0.025, 0.500]$ along with the original word2vec embedding on word similarity (left) and word analogy (right) tests. Word similarity results are presented as the average correlations from 13 different word similarity test sets.

restrictions. For POLAR, we trained two different versions. First, we obtained 1465 dimensional POLAR-large embeddings that were reported in Mathew et al. (2020), by applying polar transformation on Google's pretrained word2vec embeddings[7] using all 1465 antonym pairs. Note that these embeddings were originally trained on a much larger corpus (Google News) with a substantially larger vocabulary (3 million) than our word2vec embeddings. Therefore, POLAR-large embeddings are considerably more expensive than our imparted embeddings in terms of computational and linguistic resources. Second, we obtained 500 dimensional POLAR-small embeddings that are more comparable to imparted embeddings in terms of model dimensionality and resource usage, by performing the polar transformation on our original word2vec embeddings using the default parameters.[8] UniImp embeddings are trained on English Wikipedia (same as BiImp) using Eq. (1) ($k^g = 0.1$ as suggested in Şenel et al. (2020)) and 300 word-groups extracted from Roget's Thesaurus.

Table 2 presents interpretability scores of BiImp for $k^w \in \{0.1, 0.2, 1\}$, OIWE-IPG, SOV, Parsimax, Word2Sense, POLAR$_{small}$, POLAR$_{large}$ and UniImp along with the original word2vec embeddings in positive and negative directions separately for $n_{min} = 5$.

[7] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM.

[8] https://github.com/Sandipan99/POLAR.

Table 2
Interpretability scores (cf. Eq. (5), $n_{min} = 5$) of BiImp are higher than all baselines.

Embedding	Size	Interpretability	
		pos.	neg.
word2vec	300	12.80	12.88
OIWE-IPG	300	35.50	–
SOV	1000	14.28	13.98
Parsimax	300	18.55	17.66
Word2Sense	2250	34.11	–
POLAR$_{small}$	500	23.89	20.8
POLAR$_{large}$	1465	28.60	25.91
UniImp	300	57.49	11.38
BiImp$_{k^w=0.1}$	300	36.24	39.10
BiImp$_{k^w=0.2}$	300	42.04	46.77
BiImp$_{k^w=1}$	300	52.90	57.80

Table 3
Results on the performance evaluation tests. For BiImp, results are averaged across $k^w \in \{0.025, 0.050, \dots, 0.200\}$.

Task	w2v	IPG	SOV	Parsimax	W2S	POLAR$_s$	POLAR$_l$	UniImp	BiImp
Sem. Anlg.	79.9	32.6	52.6	79.6	12.9	70.5	60.0	80.2	79.7
Syn. Anlg.	67.6	25.6	41.6	67.5	19.4	56.1	70.8	63.4	66.3
Word Sim.	60.7	48.6	56.1	60.7	57.0	54.9	60.0	56.9	60.3
Sent. Anly.	80.3	74.5	81.8	80.3	81.2	79.1	81.8	79.0	80.00
Quest. Clf.	85.8	79.0	87.8	85.8	77.2	84.6	82.4	81.0	84.9
Sports News	95.9	95.5	96.9	96.0	86.6	94.7	91.8	96.0	95.7
Relig. News	87.0	85.8	88.6	86.9	85.1	84.1	84.9	84.9	87.4
Comp. News	81.6	78.5	86.3	81.7	73.4	77.6	72.9	80.3	80.3

Note that non-negative embeddings inherently do not have any interpretability in the negative direction. BiImp embeddings are clearly the most interpretable in the negative direction, even for small k^w ($k^w = 0.1$). For the positive direction, interpretability of BiImp is comparable with OIWE-IPG and Word2Sense and is higher than all baselines except UniImp for small k^w. For larger k^w, interpretability of BiImp is only slightly lower than that of UniImp.

5.2.2. *Preservation of semantic structure*

In addition to the intrinsic evaluation, we also evaluate the embeddings on three classification tasks:

- **Sentiment Analysis:** A sentence-level binary classification task using the Stanford Sentiment Treebank consisting of thousands of movie reviews (Socher et al., 2013) and their sentiment scores. The development and training sets in the original dataset were aggregated, and reviews with neutral scores were removed (i.e., scores between 0.4 and 0.6). The resulting dataset contained 7407 training and 1751 test samples.
- **Question Classification (TREC):** A question-level multinomial classification task using the TREC dataset (Li & Roth, 2006) consisting of six different types of questions (person, location, entity, number, description, abbreviation). This dataset consisted of 5452 training and 500 test questions.
- **News Classification:** Following Faruqui et al. (2015), three news-level binary classification tasks were considered using the 20 Newsgroup dataset.[9] The following news topics were considered (training/test sample counts): (1) Religion: atheism vs. christian (1079/716); (2) Sports: baseball vs. hockey (1192/796); (3) Computers: IBM vs. Mac (1162/775).

For these high-level NLP tasks, we took the average of the word vectors in input text (can be a sentence, question or news) as input features and trained an SVM classifier that was tuned using 5 fold cross-validation on the training sets.

Table 3 shows results. For BiImp, results are averaged across $k^w \in \{0.025, 0.050, \dots, 0.200\}$. For analogy and similarity tasks, BiImp, UniImp, Parsimax and word2vec have similar scores, suggesting that BiImp does not reduce the quality of word embeddings while improving interpretability. Both POLAR models perform slightly worse than the original embeddings (except for syntactic analogy and sentiment analysis for POLAR-large). OIWE-IPG, SOV and Word2Sense suffer from considerable performance loss in most cases, implying a reduction in the semantic information captured.

For text classification (last five lines), differences between methods are minor, except for Word2Sense embeddings, which perform poorly on question and news classification. SOV (Faruqui et al., 2015) has the best performance on classification, but recall that it has low interpretability (Table 2). BiImp performs comparably to UniImp, Parsimax and word2vec in all tasks. These results demonstrate that BiImp meets both requirements: interpretability and good task performance.

[9] http://qwone.com/~jason/20Newsgroups.

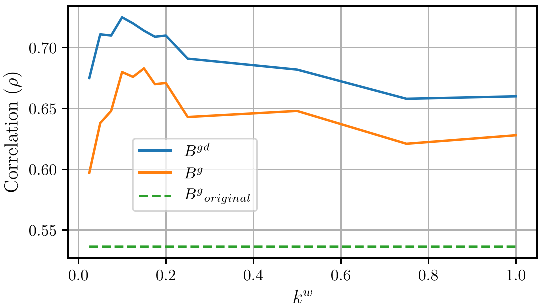


Fig. 5. Correlation of human judgments with the gender dimension in BiImp (blue, B^{gd}, Eq. (8)), with the gender vector in BiImp (orange, B^g, Eq. (7)), and with the gender vector in the original embedding space (dashed green line, $B^g_{original}$). The BiImp gender dimension clearly has the highest correlation with human judgments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

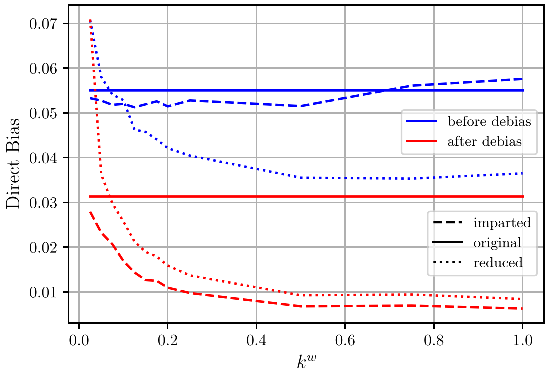


Fig. 6. Direct bias (b_1^{direct}, see Eq. (6)) of the BiImp (dashed lines) and reduced (dotted lines) embeddings as a function of k^w. Solid lines: b_1^{direct} of the original embeddings. Blue/red: Results before/after hard debiasing. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.3. Gender debiasing

5.3.1. Intrinsic bias

We calculate B^g (Eq. (7)) and B^{gd} (Eq. (8)) for BiImp. Additionally, we calculate B^g for the original word2vec embeddings ($B^g_{original}$). Fig. 5 shows that Pearson's correlation coefficients of human judgments with the BiImp gender dimension (blue, B^{gd}, Eq. (8)) is higher than their correlation with the gender vector $\vec{w}_{she} - \vec{w}_{he}$ of BiImp (orange, B^g, Eq. (7)) and also higher than the correlation with the original word2vec gender vector (dashed green line). This result suggests that BiImp's gender dimension densely captures gender information. Interestingly, B^g is much higher for BiImp than for the original embeddings ($B^g_{original}$), indicating that BiImp improves the quality of the gender vector as well.

We investigate the effect of (i) gender imparting, (ii) removing the gender dimension from the embeddings (iii) hard debiasing (Bolukbasi et al., 2016) on the gender bias level of the embedding spaces. Specifically, we measure the bias level of the original, imparted and reduced embeddings before and after hard debiasing using Eq. (6). Fig. 6 shows the bias levels. Naturally, imparting a single dimension with gender information does not alter the overall bias in the word embeddings, but rather concentrates most of the bias on a single dimension as implied by Fig. 5. Removing this dimension from the embedding space then considerably reduces the bias, especially for larger k^w. After hard debiasing, b_1^{direct} of the full and reduced imparted models (red dashed and dotted lines) are closer, and substantially lower than that of word2vec. These results show that learning an embedding space with an explicit gender dimension enhances the performance of hard debiasing.

5.3.2. Bias in classification

Prost et al. (2019) give evidence that hard debiasing introduces elevated gender bias in high-level classification tasks when compared with the original embedding model. We therefore also use *strong debiasing* (Prost et al., 2019), a method that alleviates this issue by taking N (Eq. (6)) as the entire vocabulary as opposed to just gender neutral words.

Table 4 compares original embeddings, hard debiasing, strong debiasing and the combination of BiImp and strong debiasing (B+S) on accuracy (to measure task performance) and TPR/TNR (Eq. (10), to measure classification fairness). The dataset is BiosBias. Hard debiasing has relatively high TPR/TNR, suggesting it reduces classification fairness. Strong debiasing on original word2vec

Table 4
Accuracy and True Positive/Negative Rate (TPR/TNR) on the occupation classification task. B + S = BiImp + strong debiasing.

Embedding	Acc.	TPR_{gap}	TNR_{gap}
word2vec	.717	.094	.0034
Hard debiasing	.700	.105	.0037
Strong debiasing	.699	.087	.0033
B + $\text{S}_{k^w=.1}$	.697	.066	.0022
B + $\text{S}_{k^w=.5}$	.699	.067	.0024

Table 5
Results of embeddings from gender bias experiments on the performance evaluation tests.

Task	Before debias			After debias		
	word2vec	Imparted	Reduced	word2vec	Imparted	Reduced
Sem. Anlg.	79.87	79.00 ± 0.50	79.16 ± 0.50	78.65	78.92 ± 0.57	78.99 ± 0.61
Syn. Anlg.	67.63	66.39 ± 0.99	66.48 ± 1.01	67.46	66.42 ± 0.96	66.43 ± 1.00
Word Sim.	60.68	60.08 ± 0.66	60.21 ± 0.52	60.64	60.12 ± 0.67	60.28 ± 0.53
Sent. Anly.	80.30	79.95 ± 0.36	79.94 ± 0.33	79.84	79.99 ± 0.37	79.98 ± 0.41
Quest. Clf.	85.80	84.63 ± 0.59	86.00 ± 0.92	86.20	86.27 ± 0.74	86.03 ± 0.80
Sports News	95.85	95.33 ± 0.27	95.34 ± 0.25	95.10	95.33 ± 0.27	95.33 ± 0.29
Relig. News	87.01	86.19 ± 0.61	86.10 ± 0.57	86.03	86.24 ± 0.59	86.18 ± 0.59
Comp. News	81.55	78.74 ± 0.84	78.73 ± 0.99	78.84	78.68 ± 0.83	78.63 ± 0.81

Table 6
Results of evaluation tests for the hybrid gender and interpretability imparted embeddings.

Task	$k^w = 0.1$	$k^w = 0.2$	$k^w = 1$
Semantic Anlg.	79.07	78.13	73.25
Syntactic Anlg.	66.61	65.17	45.58
Word Sim.	60.62	59.11	48.94
Sentiment Anly.	80.41	79.55	79.84
Question Clf.	84.60	85.00	84.20
Sports News	96.11	95.73	95.73
Religion News	85.89	87.43	88.55
Comput. News	81.42	81.03	79.74
$\text{Interp.}^{+}_{n_{min}=5}$	36.88	41.22	54.28
$\text{Interp.}^{-}_{n_{min}=5}$	38.47	44.79	58.50
$\text{Interp}^{+}_{n_{min}=10}$	22.41	24.17	34.07
$\text{Interp.}^{-}_{n_{min}=10}$	22.49	23.89	35.43
$\text{Gender B.}_{reduced}$	0.0470	0.0403	0.0441
$\text{Gender B.}_{debiased}$	0.0168	0.0122	0.0148

results in a relatively limited change in classification fairness. Yet when BiImp and strong debiasing are combined (B+S), TPR_{gap} and TNR_{gap} are substantially lowered without a major compromise in accuracy. These results provide further evidence that concentration of gender information on an embedding dimension improves performance of debiasing methods.

5.4. Performance of gender biased embeddings

A potential risk of debiasing on gender-imparted models is undesirable loss of semantic structure in the embedding space that might compromise task performance. To rule out this risk, we evaluate the embeddings in the gender-bias experiments on intrinsic tests and downstream classification tasks. For the imparted and reduced embeddings, we averaged the results across k^w. Table 5 shows that all the evaluated embeddings perform nearly as good as the original embeddings on all tasks, except a slightly reduced performance on computer news classification task. These results indicate that debiasing of gender-imparted embeddings successfully preserves semantic structure of the embedding space.

5.5. Hybrid gender and interpretability imparted embeddings

We demonstrate the feasibility of BiImp for concurrent gender and interpretability imparting. To do this, we obtain a hybrid model where the first dimension was encoded with gender word-groups and the remaining 299 dimensions were bidirectionally imparted with word-groups extracted from WordNet. Evaluation on gender bias, interpretability and task performance were repeated on this hybrid model. Table 6 shows the evaluation results. Hybrid model performs similarly to only WordNet imparted BiImp

(Sections 4.4 and 5.2.2) in interpretability and task performance evaluations, and performs similarly to only gender imparted BiImp (Section 5.3.1) in gender bias evaluations. These results indicate that BiImp enables gender debiasing and interpretability enhancement simultaneously in embedding models without compromising task performance.

6. Discussion of results and implications

The implications of the presented results can be organized under three main folds as follows.

- BiImp generates interpretable word embeddings by disclosing the hidden encoded structure of word embedding models without performance degradations on semantic tasks: Producing interpretable word embeddings has a critical role in deciphering the black-box behavior of language models extensively used in NLP-based information processing. Studies generating interpretable embeddings mostly give up some of the semantic properties captured by word vectors. Our experimental results show that BiImp brings interpretable word embeddings without making compromises on the semantic task performances.
- BiImp has a flexibility to be adapted to distinctive learning scenarios and semantic tasks: Aside from the main objective, BiImp is also compatible for different training schemes for word embeddings. BiImp can be easily adapted to both online learning-based and co-occurrence matrix-based training procedures. In addition, different lexical sources can be utilized without any additional cost. One can infer that BiImp presents a large spectrum of interpretable embeddings with a performance at the state-of-the-art level in various tasks ranging from word analogy to text classification.
- BiImp can also be deployed to capture and mitigate any kind of human biases that exist in word embeddings: On the other hand, imparting interpretability to word embeddings enables us to enhance word embeddings in various ways. As shown in the experimental results, capturing human biases in a dimension and removing that dimension lead to better debiasing results. This feature of BiImp embeddings can be extended to other bias types without any difficulty. Furthermore, task or domain-specific interpretable word embeddings can be obtained by adjusting the corresponding word groups assigned to embedding dimensions according to the task or domain. As a result, BiImp offers wide liberty in studying word embeddings without any further computational efforts.

7. Conclusion

We introduced BiImp, a new method for enhancing interpretability of word embeddings by bidirectional imparting of concepts extracted from lexical resources. BiImp was implemented for the scalable word2vec algorithm, and semantic concepts were extracted from Roget's Thesaurus and WordNet. In contrast to prior work, BiImp uses both directions along each dimension of the vector space separately, enabling encoding of two different concepts; the two concepts can be chosen arbitrarily or chosen as opposite concepts as a special case. As a result, BiImp makes more efficient use of the embedding space while increasing encoding flexibility.

We showed that BiImp achieves higher interpretability of word embeddings compared to state-of-the-art methods, particularly in the negative direction. At the same time, evaluation on word similarity/analogy tests as well as sentiment, news and question classification showed that BiImp does not sacrifice task performance. Thus, BiImp offers a favorable trade-off between the goals of enhancing interpretability and maintaining task performance.

BiImp represents opposite concepts in a single dimension on a continuum. As an important demonstration, we used BiImp to concentrate gender information in a single gender dimension. We showed that this gender dimension has a high correlation with stereotypical gender bias as measured by human judgments. Furthermore, we showed that this gender dimension is useful for reducing gender bias when coupled with debiasing. The combination of BiImp and debiasing achieved lower levels of gender bias and improved classification fairness. These results highlight the potential of BiImp in reducing biases and stereotypes present in word embeddings.

Here, the imparting method was demonstrated to improve interpretability and reduce gender bias in word2vec embedding models, using concepts from two common lexical sources. That said, imparting through modification of the learning objective is easily adaptable to different embedding algorithms, and to different lexical resources. The imparting framework can also be adopted for goals beyond interpretability enhancement, such as improvement of task performance. If imparting is used to encode task-relevant concepts, similar task performance can be achieved using simpler models with fewer dimensions. In turn, this can offer benefits in terms of memory requirements and computational load.

Lastly, we studied BiImp in the scope of static word embeddings. Extending BiImp to the contextualized word embeddings can be further investigated as a future work.

CRediT authorship contribution statement

Lütfi Kerem Şenel: Data curation, Software, Validation, Formal analysis, Investigation, Visualization, Resources, Writing – original draft, Writing – review & editing. **Furkan Şahinuç:** Data curation, Software, Validation, Formal analysis, Investigation, Visualization, Resources, Writing – original draft. **Veysel Yücesoy:** Validation, Writing – review & editing. **Hinrich Schütze:** Conceptualization, Methodology, Formal analysis, Supervision, Resources, Writing – review & editing, Funding acquisition. **Tolga Çukur:** Conceptualization, Methodology, Formal analysis, Supervision, Resources, Writing – original draft, Writing – review & editing, Funding acquisition. **Aykut Koç:** Conceptualization, Methodology, Formal analysis, Supervision, Resources, Writing – original draft, Writing – review & editing, Funding acquisition.

Acknowledgments

A. Koç acknowledges the support from TUBITAK 1001 Grant (120E346). T. Cukur acknowledges support from TUBA GEBIP 2015 and BAGEP 2017 fellowships.

References

Agarwal, O., Durupınar, F., Badler, N. I., & Nenkova, A. (2019). Word embeddings (also) encode human personality stereotypes. In *Proceedings of the Joint Conference on Lexical and Computational Semantics (SEM 2019)* (pp. 205–211). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/S19-1023.

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association of Computational Linguistics*, *6*, 483–495. http://dx.doi.org/10.1162/tacl_a_00034.

Bagheri, E., Ensan, F., & Al-Obeidat, F. N. (2018). Neural word and entity embeddings for ad hoc retrieval. *Information Processing & Management*, *54*, 657–673. http://dx.doi.org/10.1016/j.ipm.2018.04.007.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. http://dx.doi.org/10.1162/tacl_a_00051.

Bollegala, D., Mohammed, A., Maehara, T., & Kawarabayashi, K.-i. (2016). Joint Word Representation Learning Using a Corpus and a Semantic Lexicon. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)* (pp. 2690–2696).

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., & Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 4356–4364). Curran Associates, Inc..

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183–186. http://dx.doi.org/10.1126/science.aal4230.

Celikyilmaz, A., Hakkani-Tur, D., Pasupat, P., & Sarikaya, R. (2015). Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems. In *Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium Series*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 2180–2188). Curran Associates, Inc..

De-Arteaga, M., Romanov, A., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., & Kalai, A. T. (2019). Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *FAT* '19, Proceedings of the Conference on Fairness, Accountability, and Transparency* (pp. 120–128). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3287560.3287572.

Demszky, D., Garg, N., Voigt, R., Zou, J., Gentzkow, M., Shapiro, J., & Jurafsky, D. (2019). Analyzing polarization in social media: Method and application to tweets on 21 mass shootings. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)* (pp. 2970–3005). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1304.

Devlin, J., Chang, M.-W., Lee, K., & 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)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1423.

Dufter, P., & Schütze, H. (2019). Analytical methods for interpretable ultradense word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 1185–1191). Hong Kong, China: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D19-1111.

Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing & Management*, *57*(1), Article 102121. http://dx.doi.org/10.1016/j.ipm.2019.102121.

Fabris, A., Purpura, A., Silvello, G., & Susto, G. A. (2020). Gender stereotype reinforcement: Measuring the gender bias conveyed by ranking algorithms. *Information Processing & Management*, *57*(6), Article 102377. http://dx.doi.org/10.1016/j.ipm.2020.102377.

Faruqui, M., & Dyer, C. (2014). Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 19–24). Baltimore, Maryland: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-5004.

Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C., & Smith, N. A. (2015). Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1491–1500). Beijing, China: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P15-1144.

Fyshe, A., Talukdar, P. P., Murphy, B., & Mitchell, T. M. (2014). Interpretable semantic vectors from a joint model of brain- and text- based meaning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 489–499). Baltimore, Maryland: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-1046.

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, *115*(16), E3635–E3644. http://dx.doi.org/10.1073/pnas.1720347115.

Gonen, H., & Goldberg, Y. (2019). Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. 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)* (pp. 609–614). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1061.

Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1489–1501). Berlin, Germany: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P16-1141.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 3315–3323). Curran Associates, Inc..

Huth, A. G., de Heer, W. A., Griffiths, T. L., Theunissen, F. E., & Gallant, J. L. (2016). Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, *532*(7600), 453–458. http://dx.doi.org/10.1038/nature17637.

Iter, D., Yoon, J., & Jurafsky, D. (2018). Automatic detection of incoherent speech for diagnosing schizophrenia. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic* (pp. 136–146). New Orleans, LA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/W18-0615.

Ji, D., Gao, J., Fei, H., Teng, C., & Ren, Y. (2020). A deep neural network model for speakers coreference resolution in legal texts. *Information Processing & Management*, *57*(6), Article 102365. http://dx.doi.org/10.1016/j.ipm.2020.102365.

Ji, D., Tao, P., Fei, H., & Ren, Y. (2020). An end-to-end joint model for evidence information extraction from court record document. *Information Processing & Management*, *57*(6), Article 102305. http://dx.doi.org/10.1016/j.ipm.2020.102305.

Kocoń, J., Figas, A., Gruza, M., Puchalska, D., Kajdanowicz, T., & Kazienko, P. (2021). Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. *Information Processing & Management*, *58*(5), Article 102643. http://dx.doi.org/10.1016/j.ipm.2021.102643.

Kutuzov, A., Øvrelid, L., Szymanski, T., & Velldal, E. (2018). Diachronic word embeddings and semantic shifts: A survey. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1384–1397). Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 302–308). Baltimore, Maryland: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-2050.

Li, X., & Roth, D. (2006). Learning question classifiers: The role of semantic information. *Natural Language Engineering*, *12*(3), 229–249. http://dx.doi.org/10.1017/S1351324905003955.

Liang, P. P., Li, I. M., Zheng, E., Lim, Y. C., Salakhutdinov, R., & Morency, L.-P. (2020). Towards debiasing sentence representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 5502–5515). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.488.

Liu, Q., Jiang, H., Wei, S., Ling, Z.-H., & Hu, Y. (2015). Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1501–1511). Beijing, China: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P15-1145.

López-Santillan, R., Montes-Y-Gomez, M., Gonzalez-Gurrola, L. C., Ramirez-Alonso, G., & Prieto-Ordaz, O. (2020). Richer document embeddings for author profiling tasks based on a heuristic search. *Information Processing & Management*, *57*(4), Article 102227. http://dx.doi.org/10.1016/j.ipm.2020.102227.

Luo, H., Liu, Z., Luan, H., & Sun, M. (2015). Online learning of interpretable word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1687–1692). Lisbon, Portugal: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D15-1196.

Makhlouf, K., Zhioua, S., & Palamidessi, C. (2021). Machine learning fairness notions: Bridging the gap with real-world applications. *Information Processing & Management*, *58*(5), Article 102642. http://dx.doi.org/10.1016/j.ipm.2021.102642.

Mathew, B., Sikdar, S., Lemmerich, F., & Strohmaier, M. (2020). The POLAR framework: Polar opposites enable interpretability of pre-trained word embeddings. In *Proceedings of the Web Conference 2020* (pp. 1548–1558). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3366423.3380227.

Melchiorre, A. B., Rekabsaz, N., Parada-Cabaleiro, E., Brandl, S., Lesota, O., & Schedl, M. (2021). Investigating gender fairness of recommendation algorithms in the music domain. *Information Processing & Management*, *58*(5), Article 102666. http://dx.doi.org/10.1016/j.ipm.2021.102666.

Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–12).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 3111–3119). Curran Associates, Inc..

Miller, G. A. (1995). WordNet: A lexical database for english. *Communications of the ACM*, *38*(11), 39–41. http://dx.doi.org/10.1145/219717.219748.

Moudjari, L., Benamara, F., & Akli-Astouati, K. (2021). Multi-level embeddings for processing arabic social media contents. *Computer Speech and Language*, *70*, Article 101240. http://dx.doi.org/10.1016/j.csl.2021.101240.

Mrkšić, N., Ó Séaghdha, D., Thomson, B., Gašić, M., Rojas-Barahona, L. M., Su, P.-H., Vandyke, D., Wen, T.-H., & Young, S. (2016). Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 142–148). San Diego, California: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N16-1018.

Mumcuoğlu, E., Öztürk, C. E., Ozaktas, H. M., & Koç, A. (2021). Natural language processing in law: Prediction of outcomes in the higher courts of Turkey. *Information Processing & Management*, *58*(5), Article 102684. http://dx.doi.org/10.1016/j.ipm.2021.102684.

Murphy, B., Talukdar, P., & Mitchell, T. (2012). Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. In *Proceedings of International Conference on Computational Linguistics (COLING)* (pp. 1933–1950).

Nozza, D., Manchanda, P., Fersini, E., Palmonari, M., & Messina, E. (2021). Learningtoadapt with word embeddings: Domain adaptation of named entity recognition systems. *Information Processing & Management*, *58*(3), Article 102537. http://dx.doi.org/10.1016/j.ipm.2021.102537.

Pamungkas, E. W., Basile, V., & Patti, V. (2020). Misogyny detection in Twitter: A multilingual and cross-domain study. *Information Processing & Management*, *57*(6), Article 102360. http://dx.doi.org/10.1016/j.ipm.2020.102360.

Pamungkas, E. W., Basile, V., & Patti, V. (2021). A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing & Management*, *58*(4), Article 102544. http://dx.doi.org/10.1016/j.ipm.2021.102544.

Panigrahi, A., Simhadri, H. V., & Bhattacharyya, C. (2019). Word2Sense: Sparse interpretable word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5692–5705). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P19-1570.

Papagiannopoulou, E., & Tsoumakas, G. (2018). Local word vectors guiding keyphrase extraction. *Information Processing & Management*, *54*(6), 888–902. http://dx.doi.org/10.1016/j.ipm.2018.06.004.

Park, S., Bak, J., & Oh, A. (2017). Rotated word vector representations and their interpretability. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 401–411). Copenhagen, Denmark: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D17-1041.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/D14-1162.

Pereira, F., Lou, B., Pritchett, B., Ritter, S., Gershman, S. J., Kanwisher, N., Botvinick, M., & Fedorenko, E. (2018). Toward a universal decoder of linguistic meaning from brain activation. *Nature communications*, *9*(1), 963. http://dx.doi.org/10.1038/s41467-018-03068-4.

Pronoza, E., Panicheva, P., Koltsova, O., & Rosso, P. (2021). Detecting ethnicity-targeted hate speech in Russian social media texts. *Information Processing & Management*, *58*(6), Article 102674. http://dx.doi.org/10.1016/j.ipm.2021.102674.

Prost, F., Thain, N., & Bolukbasi, T. (2019). Debiasing embeddings for reduced gender bias in text classification. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing* (pp. 69–75). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/W19-3810.

Qian, Y., Deng, X., Ye, Q., Ma, B., & Yuan, H. (2019). On detecting business event from the headlines and leads of massive online news articles. *Information Processing & Management*, *56*(6), Article 102086. http://dx.doi.org/10.1016/j.ipm.2019.102086.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners: Technical Report*.

Roget, P. M. (2008). *Roget's International Thesaurus, 3/E*. Oxford and IBH Publishing.

Rothe, S., & Schütze, H. (2016). Word embedding calculus in meaningful ultradense subspaces. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computer Linguistics, http://dx.doi.org/10.18653/v1/p16-2083.

Roy, D., Ganguly, D., Mitra, M., & Jones, G. J. (2019). Estimating Gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. *Information Processing & Management*, *56*(3), 1026–1045. http://dx.doi.org/10.1016/j.ipm.2018.10.009.

Roy, P. K., Kumar, A., Singh, J. P., Dwivedi, Y. K., Rana, N. P., & Raman, R. (2021). Disaster related social media content processing for sustainable cities. *Sustainable Cities and Society*, *75*, Article 103363. http://dx.doi.org/10.1016/j.scs.2021.103363.

Ruan, Y.-P., Ling, Z.-H., & Hu, Y. (2016). Exploring semantic representation in brain activity using word embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 669–679). Austin, Texas: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D16-1064.

Şahinuç, F., & Koç, A. (2021). Zipfian regularities in non-point word representations. *Information Processing & Management*, *58*(3), Article 102493. http://dx.doi.org/10.1016/j.ipm.2021.102493.

Schick, T., & Schütze, H. (2020). BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 3996–4007). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.368, URL: https://aclanthology.org/2020.acl-main.368.

Şenel, L. K., Utlu, I., Şahinuç, F., Ozaktas, H. M., & Koç, A. (2020). Imparting interpretability to word embeddings while preserving semantic structure. *Natural Language Engineering*, 1–26. http://dx.doi.org/10.1017/S1351324920000315.

Şenel, L. K., Utlu, I., Yücesoy, V., Koç, A., & Cukur, T. (2018). Semantic structure and interpretability of word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(10), 1769–1779. http://dx.doi.org/10.1109/TASLP.2018.2837384.

Şenel, L. K., Yücesoy, V., Koç, A., & Cukur, T. (2018). Interpretability analysis for turkish word embeddings. In *2018 26th Signal Processing and Communications Applications Conference (SIU)* (pp. 1–4). IEEE, http://dx.doi.org/10.1109/SIU.2018.8404244.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1631–1642). Seattle, Washington, USA: Association for Computational Linguistics.

Subramanian, A., Pruthi, D., Jhamtani, H., Berg-Kirkpatrick, T., & Hovy, E. (2018). SPINE: SParse Interpretable Neural Embeddings. In: Proceedings of the Thirty Second AAAI Conference on Artificial Intelligence (AAAI).

Tuke, J., Nguyen, A., Nasim, M., Mellor, D., Wickramasinghe, A., Bean, N., & Mitchell, L. (2020). Pachinko prediction: A Bayesian method for event prediction from social media data. *Information Processing & Management*, *57*(2), Article 102147. http://dx.doi.org/10.1016/j.ipm.2019.102147.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 5998–6008). Curran Associates, Inc..

Voppel, A., de Boer, J., Brederoo, S., Schnack, H., & Sommer, I. (2021). Quantified language connectedness in schizophrenia-spectrum disorders. *Psychiatry Research*, *304*, Article 114130. http://dx.doi.org/10.1016/j.psychres.2021.114130, URL: https://www.sciencedirect.com/science/article/pii/S0165178121004261.

Yang, X., & Mao, K. (2016). Task independent fine tuning for word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(4), 885–894. http://dx.doi.org/10.1109/TASLP.2016.2644863.

Yu, M., & Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 545–550). Baltimore, Maryland: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-2089.

Yu, L.-C., Wang, J., Lai, K. R., & Zhang, X. (2017). Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(3), 671–681. http://dx.doi.org/10.1109/TASLP.2017.2788182.

Yüksel, A., Uğurlu, B., & Koç, A. (2021). Semantic change detection with gaussian word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3349–3361. http://dx.doi.org/10.1109/TASLP.2021.3120645.

Zhang, Y., Han, K., Worth, R., & Liu, Z. (2020). Connecting concepts in the brain by mapping cortical representations of semantic relations. *Nature communications*, *11*(1), 1–13. http://dx.doi.org/10.1038/s41467-020-15804-w.

Zhao, X., Wang, D., Zhao, Z., Liu, W., Lu, C., & Zhuang, F. (2021). A neural topic model with word vectors and entity vectors for short texts. *Information Processing & Management*, *58*(2), Article 102455. http://dx.doi.org/10.1016/j.ipm.2020.102455.

Zobnin, A. (2017). Rotations and interpretability of word embeddings: The case of the Russian language. In *Proceedings of the International Conference on Analysis of Images, Social Networks and Texts (AIST)* (pp. 116–128). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-73013-4_11.

Chapter 6

Does He Wink or Does He Nod? A Challenging Benchmark for Evaluating Word Understanding of Language Models

Does He Wink or Does He Nod? A Challenging Benchmark for Evaluating Word Understanding of Language Models

Lutfi Kerem Senel and **Hinrich Schütze**
Center for Information and Language Processing (CIS), LMU Munich, Germany
lksenel@gmail.com, inquiries@cislmu.org

Abstract

Recent progress in pretraining language models on large corpora has resulted in large performance gains on many NLP tasks. These large models acquire linguistic knowledge during pretraining, which helps to improve performance on downstream tasks via fine-tuning. To assess what kind of knowledge is acquired, language models are commonly probed by querying them with 'fill in the blank' style cloze questions. Existing probing datasets mainly focus on knowledge about *relations between* words and entities. We introduce WDLMPro (Word Definition Language Model Probing) to *evaluate word understanding directly* using dictionary definitions of words. In our experiments, three popular pretrained language models struggle to match words and their definitions. This indicates that they understand many words poorly and that our new probing task is a difficult challenge that could help guide research on LMs in the future.

1 Introduction

Natural language processing (NLP) has advanced drastically in the last decade with the design of larger and more sophisticated models, availability of larger corpora and increasing computational power. Pretrained word embeddings (Mikolov et al., 2013; Pennington et al., 2014) popularized the use of distributed word representations, which became a fundamental building block for NLP systems. Peters et al. (2018a) introduced LSTM-based deep contextual representations and obtained large performance gains by fine-tuning on tasks after unsupervised pretraining (Radford et al., 2018; Howard and Ruder, 2018). More recently, the attention based transformer architecture was shown to use context more effectively (Vaswani et al., 2017) and several subsequent models achieved state of the art results in many NLP tasks by combining the transformer architecture with unsupervised pretraining and task specific fine-tuning (Devlin et al., 2019; Liu et al., 2019). Radford et al. (2019) showed that language models can be applied to a variety of tasks without task specific fine tuning. This is demonstrated on a much larger scale by Brown et al. (2020).

Deep models improve performance. However, what they actually learn about language and word meaning is still to a large extent unclear due to their uninterpretable nature. For static word embeddings, researchers used word similarity (Hill et al., 2015) and word analogy (Gladkova et al., 2016) tests to shed light on what information is captured in these dense vector spaces. For language models, a great amount of linguistic knowledge is stored in the model parameters (Peters et al., 2018b). Several studies proposed using 'fill in the blank' type cloze statements to test knowledge learned by these models during unsupervised pretraining. Petroni et al. (2019) proposed the LAMA (LAnguage Model Analysis) probe to test the factual and common sense knowledge stored in language models. Similarly, Schick and Schütze (2020) introduced WNLaMPro (WordNet Language Model Probing) to assess the ability of language models to understand words based on their frequency. In WNLaMPro, cloze style questions are generated based on antonym, hypernym and cohyponym relations among words extracted from WordNet.

The existing probing datasets mainly focus on investigating the knowledge about *relations between* words or entities. However, a more direct way of testing whether a language model understands the meaning of a word is to use its dictionary definition. If a pretrained language model truly understands the meaning of a word, then it should be able to match it with its dictionary definition. Based on this motivation, we introduce the *Word Definition*

synset	definition
a_cappella_singing.n.01	*singing without instrumental accompaniment*
caroling.n.01	singing joyful religious songs (especially at Christmas)
crooning.n.01	singing in a soft low tone
singalong.n.01	informal group singing of popular songs
bel_canto.n.01	a style of operatic singing

Table 1: Five candidates from $\mathcal{G}(t)$ for t= *a_cappella_singing.n.01* and their definitions

	Noun	**Verb**
# of Synset Groups	51260	8487
Average # of Candidates	50.2	47.7
min / max # of Candidates	5 / 404	5 / 593

Table 2: WDLMPro statistics

Language Model Probing (WDLMPro) dataset;[1] it is a challenging benchmark for testing NLP models for their ability to understand words. WDLMPro is essentially a set of thousands of synset groups; each synset group consists of a target word (with its definition) and its taxonomic sisters (with their definitions). Using taxonomic sisters, rather than random word groups, makes the task more challenging for statistical models that are based on the distributional hypothesis since these words have similar distributional characteristics (Lenci, 2008). We evaluate two masked language models, BERT and RoBERTa, and the auto-regressive model GPT-2 on WDLMPro using two different probing tests: (i) match definition to word (D2W) (ii) match word to definition (W2D). We also provide a baseline using static fastText embeddings (Mikolov et al., 2018). We find that all three language models perform clearly better than the baseline. Nevertheless, they have great difficulty matching words and their definitions, implying a poor understanding of word meaning. This is an important result that could help guide research on LMs in the future.

2 WDLMPro

In this section, we introduce WDLMPro (Word Definition Language Model Probing), a dataset to test how well NLP models can match nouns and verbs with their definitions. We view this as a test of how well the models understand lexical meaning.

2.1 Dataset

WordNet (Miller, 1995) is the basis for constructing WDLMPro. A WordNet *synset* contains a set of synonyms along with a short definition of the synset. Different senses of polysemous words are represented in different synsets providing disambiguation. WordNet connects synsets with each other via semantic relations.

Based on a *target synset* t and the semantic relation hyponymy $<$, we construct a *synset group* $\mathcal{G}$ for the target as follows.

$$\mathcal{G}(t) = \{x | \exists y : t < y \land x < y\}$$

that is, G contains all synsets that are “sister hyponyms” to t with respect to a hypernym of t. $\mathcal{G}(t)$, along with the definitions of the synsets in $\mathcal{G}(t)$, will be used to set up the WDLMPro tasks that require matching of words and definitions. We discard groups $\mathcal{G}(t)$ that have a size of less than 5.

In this study, we focus on nouns and verbs, i.e., we create synset groups $\mathcal{G}$ for the nouns and verbs in WordNet. Table 1 displays five members from $\mathcal{G}(t)$ and their definitions for the target *a_cappella_singing.n.01* (see appx. for the target *beckon.v.01*.) Table 2 shows statistics of the dataset.

2.2 Probing Tests

We define two probing tests that are converses of each other:

- **Match definition to word (D2W).** Given a definition and a set of words, the task is to find the word that the definition defines.
- **Match word to definition (W2D).** Given a word and a set of definitions, the task is to find the definition that defines the word.

Each synset group $\mathcal{G}(t)$ gives rise to one instance of D2W by providing the definition of t, and all words in $\mathcal{G}(t)$. The word from $\mathcal{G}(t)$ that matches the definition has then to be identified. (Note that t is a member of $\mathcal{G}(t)$.) Similarly, each synset group $\mathcal{G}(t)$ gives rise to one instance of W2D by providing t and the definitions of all words in $\mathcal{G}(t)$. The correct definition of t has then to be identified among all definition candidates. Note that WordNet definitions by construction do not contain the word

[1] WDLAMPro and evaluation scripts are available at https://www.cis.lmu.de/definition_benchmark/WDLAMPro.zip

Masked Language Model (MLM)	
Noun	__ is <DEF>
	__ means <DEF>
	__ is defined as <DEF>
Verb	definition of __ is to <DEF>
	to <DEF> is the definition of __
Autoregressive Language Model (ALM)	
Noun	<DEF> is the definition of __
Verb	to <DEF> is the definition of __

Table 3: Patterns used for querying language models for nouns and verbs. <DEF> refers to the definition, __ is the mask or missing word that the language model has to predict.

to be defined. So there are no instances where the two tasks are trivial.

2.2.1 Application to language models

In principle, any NLP model can be tested on D2W and W2D. In this paper, we are particularly interested in testing language models. To this end, we convert the data to a format that is suitable for language models, i.e., to cloze-style questions as shown in Table 3. The basic quantity that allows us to assess the compatibility of a word t and a definition is the probability of t being generated for "__" when the definition is substituted for <DEF>.

More precisely, we compute the probability that the string representation of t is being generated. We will denote the string representation of synset t by $\boldsymbol{t}$. We obtain the string representation by removing the word type and sense information from the name of the synset and replacing underscores with white space. For example, synset *warm_up.v.04* is represented by the string "warm up".

Table 3 shows that we define different templates for masked and autoregressive language models. For the masked language models, we average the prediction scores across patterns before ranking the candidates.

2.3 Baselines

For a masked language model (MLM) M, the probability of a candidate $c \in \mathcal{G}(t)$ on W2D is calculated as:

$$P_M^{\text{W2D}}(c|t) = \prod_{i=1}^{|\boldsymbol{t}|} P(\boldsymbol{t}^i|Q(c, |\boldsymbol{t}|))$$

where $\boldsymbol{t} = [\boldsymbol{t}^1, \boldsymbol{t}^2, ..., \boldsymbol{t}^{|\boldsymbol{t}|}]$ is the tokenization produced by M. $Q(c, |\boldsymbol{t}|)$ is the input query created from one of the patterns (Table 3) with __ replaced with $|\boldsymbol{t}|$ consecutive mask tokens. For an autoregressive language model (ALM) A, we decompose $P(\boldsymbol{t}^i|Q(c), \boldsymbol{t})$ in the standard way:

$$P_A^{\text{W2D}} = \prod_{i=1}^{|\boldsymbol{t}|} P(\boldsymbol{t}^i|Q(c), \boldsymbol{t}^1, ..., \boldsymbol{t}^{i-1})$$

For D2W, we need to compare, given a definition, the probabilities of different candidate words that are generally of different lengths. To ensure a fair comparison, we follow Xiong et al. (2020). For MLMs, we match the number of mask tokens in an input query to the token count of each candidate. The final score is the average log-probability of the masked tokens:

$$P_M^{\text{D2W}}(c|t) = \frac{1}{|\boldsymbol{c}|} \sum_{i=1}^{|\boldsymbol{c}|} \log P(\boldsymbol{c}^i|Q(t, |\boldsymbol{c}|))$$

For ALMs, we use the probability of the first token:

$$P_A^{\text{D2W}}(c|t) = P(\boldsymbol{c}^1|Q(t))$$

Considering further tokens does not make sense since they are often easily predictable from the first token.

We apply our probing test to two different pretrained MLMs (BERT and RoBERTa) and one ALM (GPT-2). To investigate the effect of model size on the performance, we experiment with both base and large versions of BERT and RoBERTa along with all four sizes of GPT-2 (small, medium, large, xl). For RoBERTa, we capitalize the first letter of the candidate noun since pretrained RoBERTa models are case sensitive and expect a capital letter at the beginning of a sentence.[2]

In addition to the deep contextual language models, we also provide fastText static word embeddings[3] (Mikolov et al., 2018) as a baseline.[4] For fastText embeddings, we tokenize the candidates and their definitions using the NLTK tokenizer and represent them with their average vector. We rank candidates based on their cosine similarity to the target embedding.

[2] Not using capitalization resulted in poor performance for single token target words for D2W.

[3] We use the crawl-300d-2M-subword model from https://fasttext.cc/docs/en/english-vectors.html

[4] A reviewer suggests that it would also be interesting to investigate the performance of supervised approaches, e.g., ranking models. Our main focus here is the lexical knowledge acquired in pretraining, so we leave this for future work.

2.4 Measures

We use two measures: precision at 1 (P@1) and a rank score (RS), both based on a ranked results list, either of words or of definitions. P@1 is the percentage of top-ranked items that is correct. We define RS as follows:

$$\text{RS}(L, k) = \frac{L - k}{L - 1}$$

where $L = |\mathcal{G}(t)|$ is the number of candidates and k is the rank of the correct item, $1 \leq k \leq L$. Table 2 shows that the size of $\mathcal{G}(t)$ is highly variable; in contrast to P@1, RS is less affected by this and the random baseline (cf. Tables 4 and 5) is always 0.5.

3 Results

Tables 4 and 5 present W2D and D2W results for BERT, RoBERTa and GPT-2 along with fastText and random baselines. Language models perform clearly better than both baselines. Larger models perform generally better than smaller ones and RoBERTa consistently outperforms BERT. This might be an indication for the correlation between performance on WDLAMPro and downstream performance. However, further investigation is necessary to show the correlation more clearly. For W2D, best performance is achieved by GPT-2_{xl} for nouns (47.3 P@1, 0.81 RS) and by RoBERTa large for verbs (50.8 P@1, 0.84 RS). Performance on D2W is much lower than for W2D for all models. For nouns, RoBERTa large and GPT-2_{xl} perform similarly (28.8 and 29.8 P@1, 0.70 and 0.73 RS) while RoBERTa large achieves the best results for verbs (38.6 P@1, 0.80 RS). Poor performance on D2W compared to W2D might be due to language models' ability to distinguish different definitions better than individual words since definitions are more informative than individual words. Overall GPT-2 models perform better than masked language models (with the exception of Roberta large for verbs), despite using a single pattern as opposed to the multiple patterns used by masked language models. This might indicate that the ALM objective is better at learning word meaning than the MLM objective.

To investigate the effect of frequency, we stratify words into *rare* (fewer than 10 occurrences), *medium* (10 to 99 occurrences) and *frequent* (100 or more occurrences), based on occurrences in WWC[5] (Westbury Wikipedia Corpus, Shaoul (2010)), where we use WWC frequency as a substitute for the models' training corpora. We focus on nouns since most verbs in our dataset are relatively frequent. Table 7 shows that, for W2D, all models have a poor understanding of the meaning of rare and medium words. (See appx. for D2W results.) Even for frequent words, P@1 is never above 55.

We additionally break down the results based on the depth of the synsets in the WordNet hierarchy. Specifically, we investigate the performance of the GPT-2_{xl} model on W2D for WordNet nouns, where we take the depth of a synset group as the length of the shortest path from the target synset to the root synset (i.e., *entity.n.01*). Table 6 shows that performance drops steadily as we go deeper in the hierarchy. Lower levels of the WordNet hierarchy contain many scientific terms and names of (sub)species such as types of cattle (e.g., *cattalo*, *hereford*, *galloway*). These results suggest that even very large LMs lack the knowledge necessary to distinguish these terms.

Model	**Noun**		**Verb**	
	P@1	RS	P@1	RS
Bert$_b$	35.2	0.74	35.3	0.74
Bert$_l$	35.1	0.73	33.6	0.73
Roberta$_b$	37.1	0.75	42.7	0.79
Roberta$_l$	42.1	0.78	50.8	0.84
GPT-2_s	38.7	0.76	45.0	0.80
GPT-2_m	41.8	0.77	43.6	0.80
GPT-2_l	45.7	0.80	48.4	0.83
GPT-2_{xl}	47.3	0.81	48.6	0.83
fastText	22.5	0.66	29.1	0.69
Random	7.6	0.50	7.8	0.50

Table 4: P@1 and rank score (RS) on W2D

Model	**Noun**		**Verb**	
	P@1	RS	P@1	RS
Bert$_b$	23.7	0.65	19.3	0.65
Bert$_l$	25.4	0.65	19.3	0.65
Roberta$_b$	25.7	0.67	32.6	0.74
Roberta$_l$	28.8	0.70	38.6	0.80
GPT-2_s	23.2	0.68	29.2	0.71
GPT-2_m	25.3	0.70	27.8	0.72
GPT-2_l	28.4	0.72	31.5	0.74
GPT-2_{xl}	29.8	0.73	32.8	0.76
fastText	16.5	0.63	20.3	0.69
Random	7.6	0.50	8.0	0.50

Table 5: P@1 and rank score (RS) on D2W

[5] Targets that have more than 3 tokens (based on NLTK tokenization) are taken as rare without counting.

Depth	# synsets	# cand.	RS	P@1
3–5	2106	110	0.94	62.9
6–8	25,232	53	0.83	49.0
9–11	18,521	45	0.81	46.6
12–14	4473	19	0.74	37.4
15–19	928	13	0.67	31.5

Table 6: RS and P@1 results for GPT-2$_{xl}$ on W2D for nouns from different depths of the WordNet hierarchy. # of candidates, RS and P@1 are given as the average across all synsets within the given depth range.

Model	rare	medium	frequent	all
Bert$_b$	26.0	31.1	40.7	35.2
Bert$_l$	23.6	29.8	42.0	35.1
Roberta$_b$	30.8	34.7	40.7	37.1
Roberta$_l$	33.2	38.7	47.2	42.1
GPT-2$_s$	32.9	35.2	42.6	38.7
GPT-2$_m$	34.4	37.4	46.7	41.8
GPT-2$_l$	37.0	41.4	51.1	45.7
GPT-2$_{xl}$	37.7	42.7	53.3	47.3
Random	6.6	7.0	8.2	7.6

Table 7: P@1 scores on W2D for nouns of different frequency ranges

Analysis. The correct definition of the medium frequency verb ‘beckon’ is ‘signal with the hands or nod’. GPT-2$_{xl}$ predicts ‘signal by winking’. The correct definition of the frequent noun ‘roleplaying’ is ‘acting a particular role (as in psychotherapy)’ GPT-2$_{xl}$ predicts ‘acting the part of a character on stage’. So GPT-2$_{xl}$ understands that beckoning is signaling and that roleplaying is acting, but it has not learned to distinguish between different types of signaling and acting. This points to an important future goal for LMs: they should be developed to gain an understanding of words that goes beyond the current superficial state of the art.

Human performance on WDLAMPro. It is beyond the scope of this paper to evaluate human performance on the entirety of WDLAMPro. However, we provide a comparison with human performance on a small subset to provide an intuition about the difficulty of the task. For each of the two tasks, 20 synset groups that have a maximum of 10 candidates are randomly sampled from WDLAMPro. Then two native English speakers are asked to rank the candidates. Table 8 displays the average performance of the human participants and the language models on this subset. For both tasks, performance of the best model is comparable to the average human performance.

Model	W2D		D2W	
	P@1	RS	P@1	RS
Bert$_b$	60.0	0.84	35.0	0.64
Bert$_l$	65.0	0.74	35.0	0.69
Roberta$_b$	50.0	0.78	60.0	0.81
Roberta$_l$	55.0	0.80	45.0	0.69
GPT-2$_s$	35.0	0.69	45.0	0.71
GPT-2$_m$	50.0	0.80	50.0	0.73
GPT-2$_l$	60.0	0.84	45.0	0.75
GPT-2$_{xl}$	50.0	0.76	45.0	0.79
Human	62.5	0.88	57.5	0.77

Table 8: LM and human performance on 20 random samples of WDLAMPro.

Human performance is the upper bound for many NLP tasks. We believe that this is not the case for WDLAMPro: arguably, we should aim for models with an excellent understanding of the meanings of words even if it is better than average human understanding. Knowledge based tasks are an analogous case: we should strive for models that know as many facts as possible even if that performance is above average human performance.

4 Conclusion

We introduced WDLMPro, a probing test that helps analyze how well a model understands word meaning. WDLMPro is complementary to existing probing tests that are about *relations* between words or entities. We evaluated three popular pretrained language models on the W2D (word to definition) and D2W (definition to word) tasks. Our findings show that, despite their remarkable performance on many downstream tasks, these models struggle to match a word and its true definition, suggesting an insufficient understanding of word meaning. Relatively poor performance of these powerful models on WDLMPro can be seen as evidence for the limitations of purely distributional systems and the need for incorporating external knowledge. WDLMPro provides an important evaluation benchmark, encouraging design and training of models with precise word understanding.

Acknowledgements. We thank Denis Peskov and Sander Schulhoff for helping out with the human evaluation and the anonymous reviewers for their insightful comments and suggestions. This work was funded by the European Research Council (ERC #740516).

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Alessandro Lenci. 2008. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31.

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*.

Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proc. of the International Conference on Learning Representations (ICLR)*, pages 1–12.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *Technical Report*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8766–8774.

C. & Westbury C. Shaoul. 2010. The westbury lab wikipedia corpus.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*.

A Appendix

synset	**definition**
beckon.v.01	*signal with the hands or nod*
applaud.v.01	clap one's hands or shout after performances to indicate approval
bow.v.01	bend one's knee or body, or lower one's head
shrug.v.01	raise one's shoulders to indicate indifference or resignation
exsert.v.01	thrust or extend out
wink.v.01	signal by winking
nod.v.01	express or signify by nodding

Table 9: Seven candidates of $\mathcal{G}(t)$ for t= *beckon.v.01* and their definitions

Model	**rare**	**medium**	**frequent**	**all**
Bert$_b$	14.7	20.6	28.7	23.7
Bert$_l$	12.0	20.1	33.1	25.4
Roberta$_b$	17.7	24.2	29.5	25.7
Roberta$_l$	17.9	25.8	34.5	28.8
GPT-2$_s$	17.3	20.7	26.7	23.2
GPT-2$_m$	17.0	21.1	30.6	25.3
GPT-2$_l$	19.2	24.3	33.9	28.4
GPT-2$_{xl}$	19.3	24.8	36.3	29.8
Random	6.7	7.1	8.3	7.6

Table 10: P@1 scores on D2W for nouns based on target word frequency.

Chapter 7

CoDA21: Evaluating Language Understanding Capabilities of NLP Models With Context-Definition Alignment

CoDA21: Evaluating Language Understanding Capabilities of NLP Models With Context-Definition Alignment

Lütfi Kerem Senel, Timo Schick and **Hinrich Schütze**
Center for Information and Language Processing (CIS), LMU Munich, Germany
`lksenel@gmail.com`, `schickt@cis.lmu.de`

Abstract

Pretrained language models (PLMs) have achieved superhuman performance on many benchmarks, creating a need for harder tasks. We introduce CoDA21 (Context Definition Alignment), a challenging benchmark that measures natural language understanding (NLU) capabilities of PLMs: Given a definition and a context each for k words, but not the words themselves, the task is to align the k definitions with the k contexts. CoDA21 requires a deep understanding of contexts and definitions, including complex inference and world knowledge. We find that there is a large gap between human and PLM performance, suggesting that CoDA21 measures an aspect of NLU that is not sufficiently covered in existing benchmarks.[1]

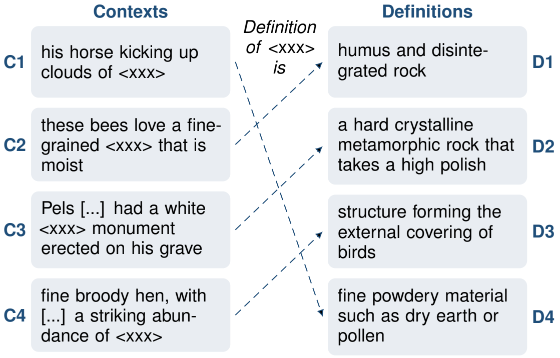

Contexts
Definitions
Definition of <xxx> is
C1
his horse kicking up clouds of <xxx>
C2
these bees love a fine-grained <xxx> that is moist
C3
Pels [...] had a white <xxx> monument erected on his grave
C4
fine broody hen, with [...] a striking abundance of <xxx>
D1
humus and disintegrated rock
D2
a hard crystalline metamorphic rock that takes a high polish
D3
structure forming the external covering of birds
D4
fine powdery material such as dry earth or pollen

Figure 1: The CoDA21 task is to find the correct alignment between contexts and definitions: **C1-D4**, **C2-D1**, **C3-D2**, **C4-D3**. The target words (**C1**:"dust", **C2**:"soil", **C3**:"marble", **C4**:"feathers"; not provided to the model) are replaced with a placeholder <xxx>.

1 Introduction

Increasing computational power along with the design and development of large and sophisticated models that can take advantage of enormous corpora has drastically advanced NLP. For many tasks, finetuning pretrained transformer-based language models (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2018) has improved the state of the art considerably. Language models acquire knowledge during pretraining that is utilized during task-specific finetuning. On benchmarks that were introduced to encourage development of models that do well on a diverse set of NLU tasks (e.g., GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019)), these models now achieve superhuman performance (He et al., 2020). The pretrain-then-finetune approach usually requires a great amount of labeled data, which is often not available or expensive to obtain, and results in specialized models that can perform well only on a single task. Recently, it was shown that generative language models can be applied to many tasks without finetuning when the task is formulated as text generation and the PLM is queried with a natural language prompt (Radford et al., 2019; Brown et al., 2020).

Motivated by recent progress in zero-shot learning with generative models as well as the need for more challenging benchmarks that test language understanding of language models, we introduce CoDA21 (**Co**ntext **D**efinition **A**lignment), a difficult benchmark that measures NLU capabilities of PLMs for the English language. Given a definition and a context each for k words, but not the words themselves, the task is to align the k definitions with the k contexts. In other words, for each definition, the context in which the defined word is most likely to occur has to be identified. This requires (i) understanding the definitions, (ii) understanding the contexts, and (iii) the ability to match the two. Since the target words are not given, a model must be able to distinguish subtle meaning differences between different contexts/definitions to be successful. To illustrate the difficulty of the task, Figure 1 shows a partial example for $k = 4$ (see Table 5 in the supplementary for the full ex-

[1] Our dataset and code are available at `https://github.com/lksenel/CoDA21`

ample). We see that both complex inference (e.g., <XXX> can give rise to a cloud by being kicked up ⇒ <XXX> must be dry ⇒ <XXX> can be dust, but not soil) and world knowledge (what materials are typical for monuments?) are required for CoDA21.

We formulate the alignment task as a text prediction task and evaluate, without finetuning, three PLMs on CoDA21: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and GPT-2 (Radford et al., 2019). Poor performance of the PLMs and a large gap between human and PLM performance suggest that CoDA21 is an important benchmark for designing models with better NLU capabilities.

2 CoDA21

2.1 Dataset

We construct CoDA21 by first deriving a set $\mathcal{G}$ of *synset groups* $\{G_1, G_2, \ldots\}$ from Wordnet (Miller, 1995). A synset group G_i is a group of synsets whose meanings are close enough to be difficult to distinguish (making the task hard), but not so close that they become indistinguishable for human and machine. In a second step, each synset group G_i is converted into a *CoDA21 group* G_i^+ – a set of triples, each consisting of the synset, its definition, and a corpus context. A CoDA21 group can be directly used for one instance of the CoDA21 task.

Synset groups. Each synset group G consists of $5 \leq k \leq 10$ synsets. To create a synset group, we start with a *parent synset* $\hat{s}$ and construct a co-hyponym group $\bar{G}(\hat{s})$ of its children:

$$\bar{G}(\hat{s}) = \{s \mid s < \hat{s}, s \notin D\}$$

where $<$ is the hyponymy relation between synsets and D is the set of synsets that have already been added to a synset group. The intuition for grouping synsets with a common parent is that words sharing a hypernym are difficult to distinguish (as opposed to randomly selected words).

We iterate $\hat{s}$ through all nouns and verbs in WordNet. At each iteration, we get all hyponyms of $\hat{s}$ that have not been previously added to a synset group; not reusing a synset ensures that different CoDA21 subtasks are not related and so no such relationships can be exploited.

We extract synset groups from co-hyponym groups by splitting them into multiple chunks of size k. In an initial exploration, we found that the task is hard to solve for human subjects if two closely related hyponyms are included, e.g., "clementine" and "tangerine". We therefore employ clustering to assemble a set of mutually dissimilar hyponyms. We first compute a sentence embedding for each hyponym definition using the *stsb-distilbert-base* Sentence Transformer model[2]. We then cluster the embeddings using complete-link clustering, combining the two most dissimilar clusters in each step. We stop merging before the biggest cluster exceeds the maximum group size ($k = 10$) or before the similarity between the last two combined clusters exceeds the maximum similarity ($\theta = 0.8$). The largest cluster G is added to the set $\mathcal{G}$ of synset groups. We then iterate the steps of (i) removing the synsets in the previous largest cluster G from $\bar{G}(\hat{s})$ and (ii) running complete-link clustering and adding the resulting largest cluster G to $\mathcal{G}$ until fewer than five synsets remain in $\bar{G}(\hat{s})$ or no cluster can be formed whose members have a similarity of less than θ.

CoDA21 groups. For each synset s, we extract its definition $d(s)$ from WordNet and a context $c(s)$ in which it occurs from SemCor[3] (Miller et al., 1994). SemCor is an English corpus tagged with WordNet senses. Let $C(s)$ be the set of contexts of s in SemCor. If $|C(s)| > 1$, we use as $c(s)$ the context in which *bert-base-uncased* predicts **s** with the highest log probability when it is masked, where **s** is the word tagged with the sense s[4] – this favors contexts that are specific to the meaning of the synset. Finally, we convert each synset group G_i in $\mathcal{G}$ to a CoDA21 group G_i^+:

$$G_i^+ = \{(s_j, d(s_j), c(s_j)) \mid s_j \in G_i\}$$

That is, a CoDA21 group G_i^+ is a set of triples of sense, definition and context. In PLM evaluation, each CoDA21 group G_i^+ gives rise to one context-definition alignment subtask.

We name the resulting dataset *CoDA21-noisy-hard*: *noisy* because if $|C(s)|$ is small, the selected context may not be informative enough to identify the matching definition; *hard* because the synsets in a CoDA21 group are taxonomic sisters, generally with similar meanings despite the clustering-based limit on definition similarity. We construct a *clean* version of the dataset by only using synsets with $|C(s)| \geq 5$. We also construct an *easy* version by

[2] https://huggingface.co/sentence-transformers/stsb-distilbert-base

[3] We do not consider synsets without contexts in SemCor.

[4] We average the probabilities when **s** is tokenized to multiple tokens.

Dataset	noun		verb	
	# of G	USC	# of G	USC
CoDA21-*clean-hard*	106	740	102	711
CoDA21-*clean-easy*	274	1999	103	758
CoDA21-*noisy-hard*	691	4633	350	2527
CoDA21-*noisy-easy*	1188	8910	370	2766

Table 1: CoDA21 group (G) statistics, USC: Unique Synset Count

taking the "hyponym grandchildren" s of a parent synset $\hat{s}$ ($s < l \wedge l < \hat{s}$) instead of its hyponym children. This reduces the similarity of synsets in a CoDA21 group, making the task easier. Table 1 gives dataset statistics.

2.2 Alignment

Recall the CoDA21 task: given a definition and a context each for k words (but not the words themselves), align the k definitions with the k contexts. That is, we are looking for a bijective function (a one-to-one correspondence) between definitions and contexts. Our motivation in designing the task is that we want a hard task (which can guide us in developing stronger natural language understanding models), but also a task that is solvable by humans. Our experience is that humans can at least partially solve the task by finding a few initial "easy" context-definition matches, removing them from the definition/context sets and then match the smaller remaining number of definitions/contexts.

The number of context-definition pairs scales quadratically ($O(k^2)$) with k and the number of alignments factorially ($O(k!)$). We restrict k to $k \leq 10$ to make sure that we do not run into computational problems and that humans do not find the task too difficult.

In order to connect contexts to definitions without using the target words, we replace the target words by a made-up word. This setup resembles the incidental vocabulary acquisition process in humans. Let t be a target word, c a context in which t occurs and m a made-up word. To test PLMs on CoDA21, we use the following pattern[5]:

$$Q(c, m) = c_m \text{ Definition of } m \text{ is}$$

where c_m is c with occurrences of t replaced by m.

We calculate the *match score* of a context-definition pair (c, d) as $\log P(d \mid Q(c, m))$, i.e., log generation probability of the definition d conditioned on $Q(c, m)$. Our objective is to maximize the sum of the k match scores in an alignment. We find the best alignment by exhaustive search. Accuracy for a CoDA21 group G_i^+ is then the accuracy of its best alignment, i.e., the number of contexts in G_i^+ that are aligned with the correct definition, divided by the total number of contexts $|G_i^+|$.

2.3 Baselines

We calculate $P(d \mid Q(c, m))$ for a masked language model (MLM) M and an autoregressive language model (ALM) A as follows:

$$P_M(d \mid Q') = \textstyle\prod_{i=1}^{|d|} P(d_i \mid Q', d_{-i})$$
$$P_A(d \mid Q') = \textstyle\prod_{i=1}^{|d|} P(d_i \mid Q', d_1, \ldots, d_{i-1})$$

where $Q' = Q(c, m)$, d_i is the i^{th} word in definition d and d_{-i} is the definition with the i^{th} word masked.

We evaluate the MLMs BERT and RoBERTa and the ALM GPT-2. We experiment with both base and large versions of BERT and RoBERTa and with all four sizes of GPT-2 (small, medium, large, xl), for a total of eight models, to investigate the effect of model size on performance.

The made-up word m should ideally be unknown so that it does not bias the PLM in any way. However, there are no truly unknown words for the models we investigate due to the word-piece tokenization they apply to the input. Any made-up word that is completely meaningless to humans will have a representation in the models' input space based on its tokenization. To minimize the risk that the meaning of the made-up word may bias the model, we use m = *bkatuhla*, a word with an empty search result on Google that most likely never appeared in the models' pretraining corpora.

In addition to PLMs, we also evaluate 2 recent sentence transformer models[6] (Reimers and Gurevych, 2019), *paraphrase-mpnet-base-v2* (mpnet) and *paraphrase-MiniLM-L6-v2* (MiniLM), and fastText static embeddings[7](Mikolov et al., 2018). To calculate the match score of a context-definition pair, we first remove the target word from the context and represent contexts and definitions as vectors. For sentence transformers, we obtain these vectors by simply encoding the input sentences. For fastText, we average the vectors of the

[5] When the target word is a verb (i.e., verb subset of a CoDA21 dataset), we add "to" at the end of our pattern.

[6] https://www.sbert.net/docs/pretrained_models.html

[7] We use the *crawl-300d-2M-subword* model from https://fasttext.cc/docs/en/english-vectors.html

words in contexts and definitions. We then calculate the match score as the cosine similarity of context and definition vectors.

3 Results

Table 2 presents average accuracy of the investigated models on the four CoDA21 datasets. As can be seen, fastText performs only slightly better than random. MLMs also perform better than random chance by only a small margin. This poor performance can be partly explained by the generation style setup we use, which is not well suited for masked language models. Even the smallest GPT-2 model performs considerably better than RoBERTa-large, the best performing MLM. Performance generally improves with model size. GPT-2_{xl} achieves the best results among the LMs on almost all datasets. Interestingly, sentence transformer *all-mpnet-base-v2* performs comparably to GPT-2_{xl} on most datasets despite its simple, similarity based matching compared to generation based matching of GPT-2 models. *Based on this observation it can be argued that current state-of-the-art language models fail to perform complex, multi-step reasoning and inference which are necessary to solve the CoDA21 tasks.* Overall, MLMs perform slightly better on verbs than nouns while the converse is true for GPT-2. As expected, all models perform better on the *easy* datasets. Performance on *noisy* and *clean* datasets are comparable; this indicates that our contexts are of high quality even for the synsets with only a few contexts.

	clean hard		clean easy		noisy hard		noisy easy		S20
Model	N	V	N	V	N	V	N	V	N
BERT$_b$	.20	.21	.22	.25	.21	.22	.22	.24	.24
BERT$_l$	.22	.22	.19	.21	.19	.20	.20	.20	.22
RoBERTa$_b$	.24	.26	.26	.32	.25	.25	.28	.27	.29
RoBERTa$_l$	.26	.30	.30	.30	.27	.29	.30	.33	.29
GPT-2$_s$	.31	.32	.42	.40	.35	.32	.40	.36	.35
GPT-2$_m$	.37	.35	.45	.39	.38	.35	.43	.39	.39
GPT-2$_l$	.38	.34	.47	**.42**	.39	**.37**	**.46**	.41	.47
GPT-2$_{xl}$	**.42**	.36	**.49**	**.42**	**.40**	.36	**.46**	**.43**	.48
mpnet	**.42**	**.39**	.48	**.42**	**.40**	**.37**	**.46**	.40	.51
MiniLM	.35	.34	.40	.36	.34	.30	.38	.32	.34
fastText	.18	.17	.20	.20	.18	.18	.18	.18	.17
Random	.15	.15	.14	.14	.16	.15	.14	.14	.14
Human	–	–	–	–	–	–	–	–	**.86**

Table 2: Average accuracy on the noun (N) and verb (V) subsets of CoDA21 for eight PLMs, two sentence transformers, fastText embeddings and (on S20) for humans

Model	bkatuhla	opyatzel	x	cloud	orange
BERT$_b$	.22	.22	.22	.23	.22
BERT$_l$	.19	.19	.20	.20	.19
RoBERTa$_b$	.26	.27	.26	.28	.28
RoBERTa$_l$	.30	.30	.29	.30	.29
GPT-2$_s$	.42	.43	.41	.39	.39
GPT-2$_m$	.45	.42	.43	.40	.41
GPT-2$_l$	.47	.46	.47	.41	.42
GPT-2$_{xl}$	.49	.44	.45	.40	.41

Table 3: Average accuracy of eight PLMs on the noun subsets of CoDA21-*clean-easy* using various words as the made-up word.

Human performance on CoDA21. We asked two NLP PhD students[8] to solve the task on S20, a random sample of size 20 from the noun part of CoDA21-*clean-easy*. Table 2 shows results on S20 for these two subjects and our models. Human performance is 0.86 – compared to 0.48 for GPT-2_{xl}, the best performing model. This difference indicates that there is a large gap in NLU competence between current language models and humans and that CoDA21 is a good benchmark to track progress on closing that gap.

To investigate the **effect of the made-up word** m, we experiment with several other words on the noun part of CoDA21-*clean-easy*. Specifically, we investigate another nonce word "opyatzel", a single letter "x" and two frequent words "orange" and "cloud". Table 3 shows the results of the models for different made-up words. MLMs do not show significant variability in performance, and perform comparably poor for all words tried. GPT2 versions, which perform considerably better than MLMs on CoDA21, perform similarly for the two nonce words and single letter "x", which do not have a strong meaning. Their performance drops significantly when the two frequent words are used as the made-up word, due to the effect of prior knowledge models have about these words.

To investigate the **effect of the pattern**, we compared our pattern $Q(c, m)$ with two alternative patterns by evaluating GPT-2_{xl} on the noun part of CoDA21-*clean-easy*. Patterns and the evaluation results are shown in Table 4. The results suggest that the effect of the pattern on performance is minimal.

Effect of the alignment setup. We constructed CoDA21 as an alignment dataset which uses the fact that matching between the definitions and contexts is one-to-one. This setup makes the task

[8] Both are proficient (though not native) English speakers.

Pattern	Acc
c_m Definition of m is	0.49
c_m m is defined as	0.51
c_m m is	0.49

Table 4: Effect of the pattern on the performance of GPT2-$_{xl}$ on the noun part of CoDA21-*clean-easy*

more intuitive and manageable for humans. However, context-definition match scores can be used to evaluate models on CoDA21 samples also without the alignment setup by simply picking context-definition pairs with the highest match score for each definition. We additionally evaluated GPT-2$_{xl}$ model on CoDA21-*clean-easy* dataset using this simple matching approach which yielded 0.38 average accuracy compared to the 0.49 accuracy achieved with the alignment setup. This result suggests that language models can also make use of the alignment style evaluation, similar to humans.

Table 5 (in the Appendix) presents a sample of size 7 from the noun part of the CoDA21-*clean-easy* dataset. Figure 2 displays all 49 match scores of the context-definition pairs for this sample obtained using GPT-2$_{xl}$. 5 of the 7 definitions (2,3,4,5,7) are matched with correct contexts with the alignment setup while 4 definitions (4,5,6,7) are matched correctly for the simple matching setup. Alignment setup enabled the model to match second and third definitions with their corresponding contexts even though their match scores are not the highest ones.

To get a better sense of why the task is hard for PLMs, we give an example, from the CoDA21 subtask in Figure 1 (also Figure 2 and Table 5 refer to the same subtask), of a context-definition match that is scored highly by GPT-2$_{xl}$, but is not correct. **Context:** "these bees love a fine-grained <XXX> that is moist". **Definition:** "fine powdery material such as dry earth or pollen". (context 6 and definition 1 in Figure 2) GPT-2$_{xl}$ most likely gives a high score because it has learned that *bees* and *pollen* are associated. It does not understand that the mutual exclusivity of "moist" and "powdery" makes this a bad match.

4 Related Work

There are many datasets (Levesque et al., 2012; Rajpurkar et al., 2016; Williams et al., 2018) for evaluating language understanding of models. Many adopt a text prediction setup: Lambada (Paperno

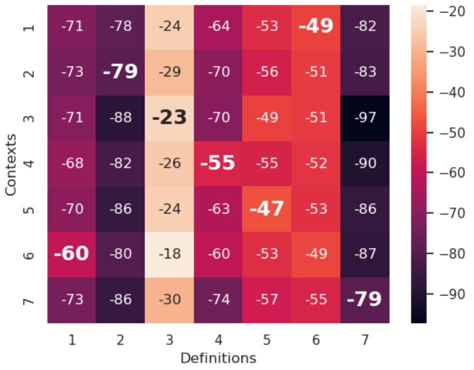


Figure 2: Match scores from GPT2-xl model for the context definition pairs for the sample given in Table 5. Match scores shown in bold correspond the context-definition pairs that are in the predicted alignment by the model that yields maximum total match score.

et al., 2016) evaluates the understanding of discourse context, StoryCloze (Mostafazadeh et al., 2016) evaluates commonsense knowledge and so does HellaSwag (Zellers et al., 2019), but examples were adversarially mined. LAMA (Petroni et al., 2019) tests the factual knowledge contained in PLMs. In contrast to this prior work, CoDA21 goes beyond prediction by requiring the matching of pieces of text. WIC (Pilehvar and Camacho-Collados, 2019) is also based on matching, but CoDA21 is more complex (multiple contexts/definitions as opposed to a single binary match decision) and is not restricted to ambiguous words. WNLaMPro (Schick and Schütze, 2020) evaluates knowledge of subordinate relationships between words, and WDLaMPro (Senel and Schütze, 2021) understanding of words using dictionary definitions. Again, matching multiple pieces of text with each other is much harder and therefore a promising task for benchmarking NLU.

5 Conclusion

We introduced CoDA21, a new challenging benchmark that tests natural language understanding capabilities of PLMs. Performing well on CoDA21 requires detailed understanding of contexts, performing complex inference and having world knowledge, which are crucial skills for NLP. All models we investigated perform clearly worse than humans, indicating a lack of these skills in the current state of the art in NLP. CoDA21 therefore is a promising benchmark for guiding the development of models with stronger NLU competence.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*. Citeseer.

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*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

George A Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. 1994. Using a semantic concordance for sense identification. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. 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.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *Technical Report*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8766–8774.

Lutfi Kerem Senel and Hinrich Schütze. 2021. Does she wink or does she nod? a challenging benchmark for evaluating word understanding of language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 532–538, Online. Association for Computational Linguistics.

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.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

A Appendices

A.1 CoDA21 group examples

Hidden word	Context
dust	1. He came spurring and whooping down the road , his horse kicking up clouds of <XXX> , shouting :
marble	2. Pels also sent a check for $ 100 to Russell 's widow and had a white <XXX> monument erected on his grave .
wastewater	3. The high cost of land and a few operational problems resulting from excessive loadings have created the need for a <XXX> treatment system with the operational characteristics of the oxidation pond but with the ability to treat more organic matter per unit volume .
feathers	4. It was a fine broody hen , white , with a maternal eye and a striking abundance of <XXX> in the under region of the abdomen .
fraction	5. It was then distilled at least three times from a trap at - 78 ‘ to a liquid air trap with only a small middle <XXX> being retained in each distillation .
soil	6. The thing is that these bees love a fine-grained <XXX> that is moist ; yet the water in the ground should not be stagnant either .
cards	7. And the coffee shop on Drexel Street , where the men spent their evenings and Sundays playing <XXX> , had a rose hedge beneath its window .
Synset	**Definition**
dust.n.01	1. fine powdery material such as dry earth or pollen that can be blown about in the air
marble.n.01	2. a hard crystalline metamorphic rock that takes a high polish; used for sculpture and as building material
effluent.n.01	3. water mixed with waste matter
feather.n.01	4. the light horny waterproof structure forming the external covering of birds
fraction.n.01	5. a component of a mixture that has been separated by a fractional process
soil.n.02	6. the part of the earth’s surface consisting of humus and disintegrated rock
card.n.01	7. one of a set of small pieces of stiff paper marked in various ways and used for playing games or for telling fortunes

Table 5: A sample CoDA21 question taken from the noun part of the CoDA21-*clean-easy* dataset. The synsets are grandchildren of the parent synset ‘material.n.01’ whose definition is “the tangible substance that goes into the makeup of a physical object”.

Hidden word	Context
suggestion	1. This was Madden 's <XXX> ; the police chief shook his head over it .
concept	2. The <XXX> of apparent black-body temperature is used to describe the radiation received from the moon and the planets .
ideals	3. Religion can summate , epitomize , relate , and conserve all the highest <XXX> and values - ethical , aesthetic , and religious - of man formed in his culture .
reaction	4. That much of what he calls folklore is the result of beliefs carefully sown among the people with the conscious aim of producing a desired mass emotional <XXX> to a particular situation or set of situations is irrelevant .
feeling	5. He had an uneasy <XXX> about it .
programs	6. The Federal program of vocational education merely provides financial aid to encourage the establishment of vocational education <XXX> in public schools .
meaning	7. Indefinite reference also carries double <XXX> where an allusion to one person or thing seems to refer to another .
theme	8. Almost nothing is said of Charles ' spectacular victories , the central <XXX> being the heroic loyalty of the Swedish people to their idolized king in misfortune and defeat .
Synset	**Definition**
suggestion.n.01	1. an idea that is suggested
concept.n.01	2. an abstract or general idea inferred or derived from specific instances
ideal.n.01	3. the idea of something that is perfect; something that one hopes to attain
reaction.n.02	4. an idea evoked by some experience
impression.n.01	5. a vague idea in which some confidence is placed
plan.n.01	6. a series of steps to be carried out or goals to be accomplished
meaning.n.02	7. the idea that is intended
theme.n.02	8. a unifying idea that is a recurrent element in literary or artistic work

Table 6: A sample CoDA21 question taken from the noun part of the CoDA21-*clean-hard* dataset. The synsets are children of the parent synset 'idea.n.01' whose definition is "the content of cognition; the main thing you are thinking about".

Bibliography

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2023. On the relation between sensitivity and accuracy in in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 155–167, Singapore. Association for Computational Linguistics.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st*

Workshop on Evaluating Vector-Space Representations for NLP, pages 1–6, Berlin, Germany. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. 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, Sashank 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. 2024. Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1).

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher R’e. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *ArXiv*, abs/2205.14135.

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.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

J. Firth. 1957. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford. Reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

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*, ICML'17, page 1243–1252. JMLR.org.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Yoav Goldberg. 2017. *Neural Network Methods for Natural Language Processing*, volume 37 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool, San Rafael, CA.

Hila Gonen, Srini Iyer, Terra Blevins, Noah Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148, Singapore. Association for Computational Linguistics.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. Book in preparation for MIT Press.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Yongchun Gu, Yi Wang, Heng-Ru Zhang, Jiao Wu, and Xingquan Gu. 2023. Enhancing text classification by graph neural networks with multi-granular topic-aware graph. *IEEE Access*, 11:20169–20183.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 1025–1035, Red Hook, NY, USA. Curran Associates Inc.

Zellig S. Harris. 1954. Distributional structure. *WORD*, 10(2-3):146–162.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Ayyoob ImaniGooghari, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

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 preprint arXiv:2310.06825.

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 preprint arXiv:2401.04088.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edition. Prentice Hall PTR, USA.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. ArXiv preprint arXiv:2001.08361.

Thomas N. Kipf and Max Welling. 2016. Variational graph auto-encoders. *CoRR*, abs/1611.07308.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Abdullatif Köksal, Timo Schick, and Hinrich Schuetze. 2023. MEAL: Stable and active learning for few-shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 506–517, Singapore. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2024. Longform: Effective instruction tuning with reverse instructions. ArXiv preprint arXiv:2304.08460.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Moxin Li, Wenjie Wang, Fuli Feng, Yixin Cao, Jizhi Zhang, and Tat-Seng Chua. 2023a. Robust prompt optimization for large language models against distribution shifts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1539–1554, Singapore. Association for Computational Linguistics.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report. ArXiv preprint arXiv:2309.05463.

Hao Liu, Matei Zaharia, and Pieter Abbeel. 2023. Ring attention with blockwise transformers for near-infinite context. ArXiv preprint arXiv:2310.01889.

Liu Liu, Zheng Qu, Zhaodong Chen, Fengbin Tu, Yufei Ding, and Yuan Xie. 2022. Dynamic sparse attention for scalable transformer acceleration. *IEEE Transactions on Computers*, 71(12):3165–3178.

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. *CoRR*, abs/1907.11692.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason. ArXiv preprint arXiv:2311.11045.

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 preprint arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1791–1799, Seattle, United States. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Nina Poerner, Benjamin Roth, and Hinrich Schütze. 2018. Interpretable textual neuron representations for NLP. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 325–327, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

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

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.

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. *J. Mach. Learn. Res.*, 21(1).

Peter Mark Roget. 2008. *Roget'S International Thesaurus, 3/E*. Oxford and IBH Publishing.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng-Xin Yong, Harshit Pandey, Michael Mckenna, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *ICLR 2022 - Tenth International Conference on Learning Representations*, Online, Unknown Region.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2022. Don't stop fine-tuning: On training regimes for few-shot cross-lingual transfer with multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10725–10742, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. ArXiv preprint arXiv:2310.11324.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Daniil Sorokin. 2021. *Knowledge Graphs and Graph Neural Networks for Semantic Parsing*. Ph.D. thesis, Technische Universität Darmstadt, Darmstadt.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Richard S. Sutton. 2019. The bitter lesson. `http://incompleteideas.net/IncIdeas/BitterLesson.html`. Accessed: 2023-09-28.

Alona Sydorova, Nina Poerner, and Benjamin Roth. 2019. Interpretable question answering on knowledge bases and text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4943–4951, Florence, Italy. Association for Computational Linguistics.

Sho Takase and Shun Kiyono. 2023. Lessons on parameter sharing across layers in transformers. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 78–90, Toronto, Canada (Hybrid). Association for 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, HyunJeong 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 Rogozińska, 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 Lučić, 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, Çağlar 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 Rakićević, 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 Kępa, 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 "Louis" Chen, 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 Rybiński, 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-Plucińska, 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, 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 preprint arXiv:2312.11805.

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. ArXiv preprint arXiv:2307.09288.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *6th International Conference on Learning Representations*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. *SuperGLUE: a stickier benchmark for general-purpose language understanding systems*. Curran Associates Inc., Red Hook, NY, USA.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. ArXiv preprint arXiv:2201.11903.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Dingjun Wu, Jing Zhang, and Xinmei Huang. 2023. Chain of thought prompting elicits knowledge augmentation. In *Findings of the Association for Computa-*

tional Linguistics: ACL 2023, pages 6519–6534, Toronto, Canada. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

W. Yu, N. Lu, X. Qi, P. Gong, and R. Xiao. 2021. Pick: Processing key information extraction from documents using improved graph learning-convolutional networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4363–4370, Los Alamitos, CA, USA. IEEE Computer Society.

Zhihang Yuan, Yuzhang Shang, Yang Zhou, Zhen Dong, Zhe Zhou, Chenhao Xue, Bingzhe Wu, Zhikai Li, Qingyi Gu, Yong Jae Lee, Yan Yan, Beidi Chen, Guangyu Sun, and Kurt Keutzer. 2024. Llm inference unveiled: Survey and roofline model insights. ArXiv preprint arXiv:2402.16363.

Chong Zhang, He Zhu, Xingyu Peng, Junran Wu, and Ke Xu. 2022. Hierarchical information matters: Text classification via tree based graph neural network. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 950–959, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. ArXiv preprint arXiv:1810.12885.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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 *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.