Efficient and Human-Inspired Natural Language Processing Methods for Multilingual and Low-Resource Settings

Ercong Nie

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

vorgelegt von
Ercong Nie
aus Henan, China

2025

Erstgutachter: PD. Dr. Helmut Schmid
Zweitgutachter: Prof. Dr. Hinrich Schütze
Drittgutachter: Prof. Dr. Zhenguang Cai

Tag der Einreichung: 24 Juni 2025
Tag der mündlichen Prüfung: 23 Oktober 2025

Abstract

The rapid advancement of large language models (LLMs) has revolutionized natural language processing (NLP), yet the benefits of these technologies remain unevenly distributed across the world's languages. Most state-of-the-art models are optimised for high-resource languages, leaving the majority of the world's linguistic diversity underrepresented and underserved. This dissertation addresses the dual challenge of efficiency and inclusivity in multilingual NLP by developing novel, human-inspired methods that extend the reach of language technology to low-resource settings.

The research is organised around four interrelated threads. First, the dissertation advances **prompt-based learning for multilingual prediction**, introducing robust calibration techniques and cross-lingual retrieval-augmented prompting (PARC) to mitigate label bias and enhance zero- and few-shot performance, particularly for low-resource and typologically diverse languages. Apart from classification tasks, we extend the applications of prompt-based learning to other multilingual task types. Decomposed prompting strategies are proposed to probe the linguistic structure knowledge encoded in LLMs, while the BMIKE-53 benchmark extends prompt-based learning to cross-lingual knowledge editing, enabling systematic evaluation across 53 languages.

Second, the work systematically investigates **prompt-based fine-tuning for zero-shot cross-lingual transfer**. Through comprehensive empirical studies, it is demonstrated that prompt-based fine-tuning consistently outperforms traditional approaches for both classification and structured prediction tasks, including part-of-speech tagging and named entity recognition. Beyond modern languages, we exemplify the application of cross-lingual transfer to historical language processing by applying a delexicalized constituency parser for Middle High German.

Third, the dissertation addresses practical constraints in low-resource NLP deployment by introducing **efficient data augmentation and parameter-efficient fine-tuning methods**. The AMD^{2}G framework enables robust multi-domain dialogue generation in low-resource settings through domain-agnostic training and adaptation, while GNNavi leverages graph neural networks to guide information flow in prompt-based fine-tuning, achieving competitive results with minimal parameter updates.

Finally, the dissertation shifts focus to **human-inspired interpretability and mechanistic understanding of language models**. By integrating psycholinguistic and neurolinguistic probing paradigms, it reveals a persistent gap between model performance and true linguistic competence, with LLMs demonstrating stronger mastery of linguistic form than meaning. Mechanistic interpretability techniques are employed to trace and mitigate language confusion in English-centric LLMs, showing that targeted neuron-level interventions can robustly improve multilingual reliability without sacrificing general competence.

Collectively, these contributions advance both the practical capabilities and scientific understanding of multilingual NLP. The dissertation demonstrates that prompt-based and parameter-efficient methods, when combined with human-inspired analysis, can make NLP more inclusive, interpretable, and robust. Looking forward, the work highlights the importance of developing culturally and socially aware language models, extending interpretability research to cross-cultural conceptual understanding, and leveraging insights from cognitive science and neuroscience to inspire the next generation of human-centric multilingual NLP systems.

Zusammenfassung

Der rasante Fortschritt großer Sprachmodelle (Large Language Models, LLMs) hat das Natural Language Processing (NLP) grundlegend revolutioniert. Dennoch sind die Vorteile dieser Technologien weltweit ungleich verteilt: Die meisten modernen Modelle sind auf ressourcenstarke Sprachen optimiert, wodurch der Großteil der sprachlichen Vielfalt unterrepräsentiert und unzureichend unterstützt bleibt. Diese Dissertation adressiert die doppelte Herausforderung von Effizienz und Inklusivität im mehrsprachigen NLP, indem neuartige, menscheninspirierte Methoden entwickelt werden, die den Einsatz von Sprachtechnologie auf ressourcenarme Kontexte ausweiten.

Die Forschung ist um vier miteinander verbundene Schwerpunkte organisiert. Erstens werden **promptbasierte Lernverfahren für mehrsprachige Vorhersagen** weiterentwickelt. Robuste Kalibrierungstechniken und sprachübergreifende, retrieval-erweiterte Prompts (PARC) werden eingeführt, um Label-Bias zu mindern und die Zero- und Few-Shot-Performance insbesondere für ressourcenarme und typologisch diverse Sprachen zu verbessern. Über Klassifikationsaufgaben hinaus werden promptbasierte Methoden auf weitere mehrsprachige Aufgaben ausgeweitet. Problem-zerlegende Prompting-Strategien dienen dazu, das in LLMs kodierte Wissen über linguistische Strukturen gezielt zu untersuchen, während der BMIKE-53-Benchmark promptbasiertes Lernen auf das cross-linguale Knowledge Editing ausweitet und eine systematische Evaluation in 53 Sprachen ermöglicht.

Zweitens wird **promptbasiertes Fine-Tuning für Zero-Shot-Cross-Lingual-Transfer** systematisch untersucht. Unsere umfangreichen empirischen Studien zeigen, dass promptbasiertes Fine-Tuning traditionelle Ansätze sowohl bei Klassifikations- als auch bei strukturierten Vorhersageaufgaben (z.B. POS-Tagging, Named Entity Recognition) konsistent übertrifft. Über moderne Sprachen hinaus veranschaulichen wir die Anwendung des cross-lingualen Transfers auf die Verarbeitung historischer Sprachen, indem wir einen delexikalisierten Konstituentenparser für Mittelhochdeutsch anwenden.

Drittens werden praxisnahe Einschränkungen beim Einsatz von NLP in ressourcenarmen Umgebungen adressiert, indem **effiziente Methoden zur Datenaugmentation und zum parameter-effizienten Fine-Tuning** eingeführt werden. Das AMD^{2}G-Framework ermöglicht robuste, mehr-Domänen-Dialoggenerierung in ressourcenarmen Umgebungen durch domänen-unabhängiges Training und Adaption, während GNNavi Graph-Neural-Networks nutzt, um den Informationsfluss beim promptbasierten Fine-Tuning gezielt zu steuern und mit minimalen Parameteranpassungen wettbewerbsfähige Ergebnisse zu erzielen.

Schließlich richtet die Dissertation den Fokus auf **menscheninspirierte Interpretierbarkeit**

und mechanistisches Verständnis von Sprachmodellen. Durch die Integration psycholinguistischer und neurolinguistischer Probing-Paradigmen wird eine beständige Lücke zwischen Mo-dellleistung und tatsächlicher Sprachkompetenz aufgezeigt. Zudem wird festgestellt, dass LLMs die linguistische Form besser beherrschen als die Bedeutung. Mechanistische Interpretierbarkeitsverfahren werden eingesetzt, um Sprachverwirrung in englischzentrierten LLMs zu analysieren und zu beheben; gezielte Interventionen auf Ebene der Neuronen verbessern dabei die Zuverlässigkeit bei anderen Sprachen, ohne die allgemeine Kompetenz zu beeinträchtigen.

In ihrer Gesamtheit erweitert diese Dissertation sowohl die praktischen Möglichkeiten als auch das wissenschaftliche Verständnis der multilingualen Sprachverarbeitung. Die Dissertation zeigt, dass promptbasierte und parameter-effiziente Methoden, kombiniert mit mencheninspirierter Analyse, NLP inklusiver, interpretierbarer und robuster machen können. Für die Zukunft wird die Bedeutung der kulturellen und sozialen Sensibilität der Sprachmodelle hervorgehoben, die Erweiterung der Interpretierbarkeitsforschung auf kulturübergreifende konzeptuelle Repräsentationen angeregt und das Potenzial interdisziplinärer Ansätze aus Kognitionswissenschaft und Neurowissenschaft für die nächste Generation menschenzentrierter, mehrsprachiger NLP-Systeme betont.

Declaration on Writing Aids with AI Tools

In the preparation of this dissertation, artificial intelligence (AI) tools, specifically ChatGPT, have been employed as writing aids. The use of these tools has been conducted in accordance with academic integrity guidelines and with full transparency regarding their role in the writing process. The following summarizes the specific ways in which AI tools have contributed to the composition of this dissertation:

Writing Refinement Across all chapters, ChatGPT has been utilized to assist in refining grammar and writing style. This includes the identification and correction of grammatical errors, the rephrasing of sentences to avoid unnatural or ambiguous expressions, and the improvement of word choice, including the selection of more precise and descriptive terminology where appropriate. All suggestions and refinements provided by ChatGPT for the purpose of writing improvement have been carefully and critically reviewed by the author to ensure accuracy and to maintain alignment with the original meaning and intent of the text.

Literature Suggestions On rare occasions, a specialized version of GPT (Scholar GPT) was used to obtain suggestions for relevant literature in specific research domains, most notably in Chapter 2. All recommended publications were subsequently checked manually by the author to verify their correctness, relevance, and suitability for inclusion in the dissertation.

The use of AI tools in this dissertation has been limited to the above-mentioned supportive functions. At no point were AI tools used to generate original research content, analyze data, or draw scientific conclusions. The author remains solely responsible for the originality, accuracy, and scholarly integrity of all substantive content presented herein.

Contents

Chapter 1

Introduction

In the digital age, language technology has become a transformative force, shaping societies, economies, and cultures worldwide. Natural language processing (NLP)—the science and engineering of computationally understanding and generating human language—now powers search engines, digital assistants, translation platforms, educational tools, and critical information systems across the globe. Despite these advances, the practical and scientific reach of NLP remains uneven. A major catalyst for recent progress in NLP has been the development of large language models (LLMs). These models have brought unprecedented advances, endowing systems with remarkable fluency, generalization, and adaptability. However, while LLMs have revolutionized the field, significant challenges persist—particularly in making these models efficient, robust, and aligned with human needs. This is especially true for multilingual and low-resource scenarios, where the majority of the world's linguistic diversity remains underrepresented and underserved. Thus, the promise of language technology is tempered by the need to ensure that its benefits are accessible, equitable, and scientifically grounded for all languages and communities. Addressing these challenges is central to the research presented in this dissertation.

On the one hand, the progress of language technology is unequally distributed: their remarkable fluency and reasoning abilities are disproportionately concentrated in a handful of high-resource, predominantly English-speaking domains; the vast majority of the world's languages and domains remain underserved, their speakers and practitioners constrained by the data- and resource-hungry character of state-of-the-art models. This disparity is stark: thousands of the world's languages, and billions of their speakers, remain marginalized by data scarcity, resource imbalance, and technological ignorance. On the other hand, the scientific underpinnings of LLMs—their internal mechanisms, cognitive plausibility, and true linguistic competence—are only partially understood, raising foundational questions about what it means for machines to "understand" language. To answer these questions about the internal workings, transparency, and reliability of these models is essential for the road to Artificial General Intelligence (AGI). Understanding the working mechanisms of LLMs in processing human language text from a human-inspired perspective requires particular attention when applied beyond the familiar terrain of English and well-resourced tasks.

Against this backdrop, multilingual NLP emerges not only as a grand challenge for artificial intelligence but as a scientific imperative: it is central to ensuring equity, access, and robust performance in global language technologies. Moreover, research that is human-inspired—not merely engineering-oriented—pushes the field toward models and methods that reflect the flexibility, adaptability, and interpretability of human cognition. This dissertation is positioned at the intersection of these urgent needs. In response to these intertwined scientific and practical challenges, this dissertation charts a path toward **efficient and human-inspired NLP for multilingual and low-resource settings**. It advances the field through the development of efficient and human-inspired NLP **methods** for multilingual and low-resource settings, integrating algorithmic innovation with cognitive and mechanistic insight.

A core methodology employed in this dissertation is **prompt-based learning (e.g., in-context learning)**, which leverages language models to directly predict the probability of text for various NLP tasks. To enable this, the input is reformulated into a cloze-style prompt, allowing the language model to generate the desired output on the masked token position or the next token position. For example, in sentiment analysis, the input "This product is amazing" is transformed into "This product is amazing. In summary, it is a `[MASK]` product". The model then predicts the missing word, such as "great" or "terrible", to determine the sentiment. However, a key problem with prompt-based learning is the **bias in masked token prediction**, where models favor label words that frequently occurred during pretraining. To address this, this dissertation introduces **calibration techniques** that modify the probabilities of label words predicted by the models, leading to substantial performance gains, particularly in multilingual settings. Besides using calibration techniques, we enhance prompts with cross-lingual retrieval. We propose the **PARC** (Prompt Augmented by Cross-Lingual Retrieval) pipeline to enhance the multilingual prediction performance for low-resource languages, in multilingual tasks such as topic classification, natural language inference, paraphrase detection, etc. Many languages lack annotated data for fine-tuning for these tasks. To address this, the PARC method is proposed, which augments prompts for low-resource languages by retrieving semantically similar examples from high-resource language corpora. This cross-lingual retrieval-augmented prompting enables better zero-shot learning and bridges the performance gap between high- and low-resource languages.

Another key methodology is **cross-lingual transfer learning**, which leverages knowledge from high-resource languages to adapt models to low-resource settings. This dissertation investigates prompt-based fine-tuning for zero-shot cross-lingual transfer. Different from vanilla fine-tuning, which mostly relies on the `CLS` token to map an input sentence to a label ID, prompt-based fine-tuning utilizes a prompt template to map the mask token or the next token to a label word, which is more aligned with the language modeling in the pretraining and captures more contextual information during the fine-tuning. This dissertation introduces **PROFIT**, a pipeline that systematically compares prompt-based and vanilla fine-tuning for cross-lingual transfer, and demonstrates that prompt-based fine-tuning consistently outperforms vanilla fine-tuning, especially in few-shot scenarios. However, the effectiveness of cross-lingual transfer is constrained by linguistic similarity and the uneven representation of languages in pre-training corpora. To extend the applications of prompt-based fine-tuning to more task types, such as sequence labeling, the dissertation proposes **ToPro**, a token-level prompt decomposition method for cross-lingual structured prediction, such as part-of-speech (POS) tagging and named entity recognition (NER). ToPro achieves state-of-the-art performance, particularly for languages that are typologically distant from English.

Furthermore, the dissertation addresses the challenge of **efficient NLP in low-resource settings** through data augmentation and parameter-efficient fine-tuning. It introduces **AMD^{2}G**, a unified data augmentation framework for low-resource multi-domain dialogue generation. By decoupling domain-agnostic and domain-specific features through de-domaining and sequential training, AMD^{2}G achieves superior performance across multiple domains. The dissertation also introduces **GNNavi**, a parameter-efficient fine-tuning method that integrates a graph neural network layer into large language models to guide information aggregation in prompt-based fine-tuning, achieving superior performance with minimal parameter updates.

Finally, the dissertation explores the **interpretability of LLMs** through human-inspired probing methods and mechanistic analysis. It employs minimal pair probing to distinguish between form and meaning representations, revealing that LLMs encode form more robustly than meaning. It also applies mechanistic interpretability to diagnose and address language confusion, identifying critical neurons and proposing editing strategies to improve multilingual reliability.

By tackling these core challenges, such as prompt-based learning biases, cross-lingual transfer, parameter-efficient adaptation, and model interpretability, this work aims not only to broaden the practical impact of NLP, but also to deepen our understanding of the principles that could inspire the next generation of inclusive and reliable language technologies. In doing so, this thesis aspires to contribute both foundational knowledge and practical methods to realize NLP that is globally relevant, scientifically sound, and inspired by the very nature of human language.

This dissertation follows a scientific journey starting from the limitations of English-centric, resource-hungry NLP models, through the design of efficient, cognitively-inspired algorithms, to a mechanistic and human-centric understanding of language models. The research of this dissertation is unified by a dual focus: (1) Efficiency—developing methods that make NLP viable for low-resource and multilingual contexts via prompt-based methods, robust transfer, and parameter-efficient fine-tuning; and (2) Human-Inspiration—drawing on psycholinguistics and mechanistic interpretability to understand and improve model generalization, reliability, and inclusivity.

The research traverses several interlocking themes:

- First, it identifies and mitigates the biases and inefficiencies of prompt-based learning for multilingual prediction, introducing calibration and retrieval-augmented methods to robustly bridge high-resource and low-resource languages.

- Second, it extends prompt-based approaches to fine-tuning and structured prediction, proposing decomposed and token-level prompt strategies that enable robust cross-lingual transfer even in challenging sequence labeling and parsing tasks, including for historical languages.

- Third, it addresses efficiency through data augmentation and parameter-efficient fine-tuning, leveraging graph neural networks and unified data augmentation strategies to maximize impact in low-resource dialogue and classification settings.

- Finally, it closes the loop by turning an interpretability lens on language models, exploring their linguistic competence versus performance, probing internal mechanisms of language confusion, and developing targeted model editing to mitigate failures.

The result is a theoretically grounded, empirically validated, and human-aligned set of methods for multilingual and low-resource NLP. It follows a path from practical bottlenecks to scientific insight and back, advancing both the art and science of language technology.

1.1 Research Motivation

1.1.1 Language Inequality and the Value of Multilingual NLP

NLP technologies have become foundational to modern society, powering applications from search engines and digital assistants to translation services and content moderation. The Transformer architecture (Vaswani et al., 2017) and the rise of large pre-trained language models (PLMs) such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), and GPT-3 (Brown et al., 2020) have driven remarkable advances in language understanding and generation. Multilingual PLMs (MPLMs) like mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), and mT5 (Xue et al., 2021) have extended these capabilities to over 100 languages.

Despite these advances, a profound gap persists: the vast majority of the world's 7,000+ languages (Joshi et al., 2020) remain underrepresented or entirely excluded from digital NLP tools. Most MPLMs cover only a fraction of global linguistic diversity, and even within covered languages, performance is highly uneven (Wu and Dredze, 2020; Etxaniz et al., 2024). English and a handful of high-resource languages dominate both training data and model performance, while low-resource and endangered languages are left behind. This digital language divide is not merely a technical issue—it has deep implications for information access, social equity, and the preservation of cultural heritage. Speakers of marginalized languages are often denied access to essential technologies, from hate speech detection to information retrieval, exacerbating global inequalities.

The root cause of this inequality is data scarcity. While massive text corpora exist for English and a few other languages, most languages lack the annotated or even raw data necessary for training modern NLP systems. As a result, the benefits of NLP remain concentrated among speakers of high-resource languages, while billions are excluded. Addressing this challenge is both a scientific and ethical imperative: multilingual NLP research is essential for building inclusive, equitable, and globally relevant language technologies.

1.1.2 Benefits of Cross-Lingual Transfer Learning

The paradigm of pre-training and fine-tuning has revolutionized NLP, but it is fundamentally limited by the availability of large-scale data and computational resources. For low-resource languages, the lack of data makes it infeasible to train dedicated PLMs, and the environmental cost of large-scale pre-training is increasingly unsustainable. Cross-lingual transfer learning has emerged as a key strategy: by leveraging knowledge from high-resource languages, models can be adapted to low-resource settings with minimal or no labeled data (Hu et al., 2020b; Gao et al., 2021; Lin et al., 2022).

Early approaches to cross-lingual transfer relied on parallel word embeddings (Mikolov et al., 2013b; Gouws and Søgaard, 2015), while modern MPLMs share parameters across languages to enable zero-shot and few-shot transfer. However, the effectiveness of cross-lingual transfer is constrained by linguistic similarity, script differences, and the uneven representation of languages in pre-training corpora (Lauscher et al., 2020; Chang et al., 2022). Many languages remain unseen or underrepresented, and transfer performance drops sharply for typologically distant or unseen languages.

These challenges are even more acute in the context of historical and ancient languages, such as Middle High German (MHG). For such languages, the scarcity of digital resources is compounded by the lack of annotated corpora and the high cost of expert annotation (Nie et al., 2023a). Yet, historical languages are of immense value for linguistic research, cultural heritage, and the study of language evolution. Cross-lingual transfer techniques, especially those leveraging structural similarities between historical and modern languages, offer a promising solution. For example, delexicalized constituency parsing can exploit the syntactic continuity between MHG and Modern German (MG), enabling robust syntactic analysis even in the absence of annotated treebanks for the historical language (Hirschmann and Linde, 2023).

Thus, advancing cross-lingual transfer not only addresses the needs of contemporary low-resource languages but also opens new avenues for computational historical linguistics, supporting the automatic analysis and preservation of ancient texts.

1.1.3 Efficiency and Adaptation for Low-Resource NLP

The remarkable performance of LLMs has come at the cost of enormous computational resources and environmental impact. Training and deploying these models for every language and domain is impractical and unsustainable, especially as public data becomes exhausted and privacy concerns mount. Many real-world scenarios—such as dialogue generation in specialized domains or adaptation to new languages—require efficient learning from limited data or with limited computing resources.

Parameter-efficient fine-tuning (PEFT) (Hu et al., 2022) and data augmentation (Sennrich et al., 2016a) have emerged as promising approaches. These methods seek to maximize the impact of available data and minimize the number of parameters that must be updated. Integrating these approaches with prompt-based and cross-lingual learning paradigms opens new possibilities for scalable, fair, and privacy-preserving NLP.

1.1.4 Human-Inspired and Interpretable NLP: Beyond Black-Box Performance

While large language models have achieved impressive results, fundamental questions remain about their true linguistic competence, cognitive plausibility, and reliability. Models often succeed by exploiting statistical regularities rather than genuine understanding, leading to brittle behavior, hallucinations, and failures in multilingual or low-resource contexts (Bender and Koller, 2020). The gap between observed performance and underlying competence is especially pronounced when models are evaluated outside their training distribution or asked to generalize to new languages and tasks.

Human-inspired NLP seeks to bridge this gap by drawing on insights from psycholinguistics, neurolinguistics, and cognitive science. Mechanistic interpretability—analyzing the internal computations and representations of LLMs—offers new tools for understanding the gap between performance and competence (Elhage et al., 2021). Probing methods, minimal pair analysis, and neuron-level interventions can reveal how models encode linguistic structure, conceptual knowledge, and cross-lingual transferability (Belinkov and Glass, 2019). Developing interpretable, human-aligned models is not only a scientific goal but also a practical necessity for building trustworthy, robust, and inclusive NLP systems.

1.1.5 A Unified Vision: From Bottlenecks to Scientific Insight

Bringing together these threads, this dissertation is motivated by the urgent need for **efficient and human-inspired NLP methods** that extend the reach of language technology to **multilingual** and **low-resource settings**, while advancing our scientific understanding of language models. By addressing practical bottlenecks—such as bias in prompt-based learning, data scarcity, and language confusion—and grounding solutions in cognitive and mechanistic principles, this work aspires to contribute both new methods and new insights to the field.

1.2 Research Questions

The research presented in this dissertation is unified by the overarching goal of advancing efficient and human-inspired NLP for multilingual and low-resource settings. To address this goal, the dissertation is structured around **four** interrelated thematic groups of research questions, each reflecting a critical aspect of the field. Each thematic group forms the focus of an individual chapter.

The first group, ***Prompt-Based Learning for Multilingual Prediction*** (Chapter 3), focuses on overcoming the limitations of current prompt-based learning methods for multilingual and low-resource languages. Here, the research questions investigate how to calibrate and augment prompt-based learning and how to expand prompt-based learning for more various multilingual task types. It is worth noting that this group of research questions focuses on training-free prompt-based learning methods, that is to say, parameter updating via fine-tuning is not involved in this context.

The second group, ***Prompt-Based Fine-Tuning for Zero-Shot Cross-Lingual Transfer*** (Chapter 4), involves the fine-tuning part of prompt-based learning for multilingual tasks with the zero-shot cross-lingual transfer paradigm. In this transfer learning paradigm, the multilingual models are only fine-tuned on the source language and then directly evaluated on the target language samples. This group of questions investigates the advantages of prompt-based fine-tuning over vanilla fine-tuning across various multilingual task types and explores how to enable effective cross-lingual transfer for historical languages.

The third group, ***Efficient NLP Methods for Low-Resource Settings*** (Chapter 5), addresses the practical challenges of deploying NLP in resource-constrained scenarios, i.e., where the data resources or computing resources are limited. This includes questions on unified data augmentation and parameter-efficient fine-tuning, with the aim of maximizing performance across diverse domains.

The fourth group, ***Human-Inspired Analysis and Interpretability of Language Models*** (Chapter 6), seeks to bridge the gap between model performance and true linguistic competence. These questions probe the internal mechanisms of large language models, explore the relationship between form and meaning, and develop mechanistic and neuron-level interventions to mitigate failure modes such as language confusion. This group also considers how insights from cognitive science and mechanistic interpretability can inform the design of more reliable, interpretable, and human-aligned NLP systems.

The specific research questions of each group guiding this dissertation are as follows:

1. Prompt-Based Learning for Multilingual Prediction

i. **Prompt Calibration and Augmentation (§3.1 and §3.2):** How can prompt-based learning methods be calibrated and augmented to overcome inherent biases and improve zero-shot and few-shot multilingual prediction, particularly for low-resource and typologically diverse languages?
This question seeks to address the limitations of prompt-based approaches, such as label bias and insufficient cross-lingual generalization, by exploring probability calibration, retrieval-augmented prompting, and the integration of cross-lingual information to enhance the robustness and inclusivity of multilingual NLP systems.

ii. **Prompt-based Learning for Multilingual Sequence Labeling (§3.3):** What strategies enable robust and efficient prompt-based learning for multilingual sequence labeling tasks across diverse languages, and how can these methods evaluate and probe linguistic knowledge in large language models?
Here, the focus is on extending prompt-based methods to sequence labeling tasks, introducing decomposed prompting to probe the multilingual capabilities of both English-centric and multilingual LLMs.

iii. **Cross-Lingual Knowledge Editing (§3.4):** How can in-context learning and prompt-based approaches be leveraged for cross-lingual knowledge editing, and what are the key factors affecting their reliability and generalization across languages?
This question investigates the potential and limitations of in-context learning for knowledge editing in a multilingual context, analyzing the impact of model scale, demonstration design, and linguistic properties on the effectiveness of knowledge updates across a wide range of languages.

2. Prompt-Based Fine-Tuning for Zero-Shot Cross-Lingual Transfer

i. **Prompt-Based Fine-Tuning for Cross-Lingual Transfer (§4.1):** Does prompt-based fine-tuning consistently outperform vanilla fine-tuning for zero-shot cross-lingual transfer, and what factors govern its effectiveness across tasks, languages, and resource conditions? This question systematically compares prompt-based and vanilla fine-tuning for cross-lingual transfer, analyzing performance trends, task types, and the influence of language similarity and pretraining data size, with the goal of identifying best practices for efficient and scalable adaptation.

ii. **Prompt-Based Fine-Tuning for Structured Prediction (§4.2) and §4.3):** What strategies extend prompt-based fine-tuning for structured prediction tasks, such as part-of-speech and parsing, across diverse languages, including historical languages?
Here, the focus is to propose new strategies to extend prompt-based fine-tuning to token-level and structured tasks within the zero-shot cross-lingual transfer paradigm, introducing token-level prompting and delexicalization to improve the cross-lingual transfer performance of multilingual models with limited resources, including historical languages.

3. Efficient NLP Methods for Low-Resource Settings

i. **Data Augmentation and Parameter-Efficient Adaptation (§5.1 and §5.2):** What data augmentation and parameter-efficient fine-tuning strategies can enhance low-resource NLP tasks, such as multi-domain dialogue generation and classification, and how does domain or task similarity affect transferability and performance?
This question explores the design and evaluation of data augmentation frameworks and parameter-efficient adaptation methods, such as GNN-based fine-tuning, to maximize the impact of limited data and computational resources in low-resource data and computing scenarios.

4. Human-Inspired Analysis and Interpretability of Language Models

i. **Human-Inspired Probing Methods (§6.1):** What is the relationship between performance and competence in large language models, and how can human-inspired probing paradigms reveal the internal representations of form and meaning across languages?
This question examines the gap between observed performance and underlying linguistic competence in LLMs, using minimal pair probing to distinguish between form and meaning representations, and to assess the cognitive plausibility of model behavior.

ii. **Mechanistic Understanding of Language Confusion (§6.2):** When LLMs answer a question in the wrong language, can we trace back this failure to a few specific neurons and prevent the failure by manipulating these neurons without harming general competence?
This question applies mechanistic interpretability to diagnose and address language confusion, identifying critical neurons and proposing editing strategies to improve multilingual reliability and interpretability, thereby advancing the scientific understanding and practical robustness of language models.

1.3 Research Contributions

The work in this dissertation makes substantial contributions to the advancement of efficient and human-inspired NLP for multilingual and low-resource settings. Addressing the research questions outlined in Section §1.2, the contributions are organized into four interrelated areas, the same as the topics of the research question groups: (1) prompt-based learning for multilingual prediction, (2) prompt-based fine-tuning for zero-shot cross-lingual transfer, (3) efficient NLP methods for low-resource settings, and (4) human-inspired analysis and interpretability of language models. Together, these contributions form a coherent progression from foundational methods to scientific insight and are summarized as follows:

1. **Prompt-Based Learning for Multilingual Prediction**

- *Calibration and Augmentation for Prompt-Based Learning:* This dissertation introduces novel probability calibration techniques and retrieval-augmented prompting pipelines to address the inherent biases and limitations of prompt-based learning in multilingual and low-resource contexts. By analyzing and mitigating label bias and by leveraging cross-lingual retrieval from high-resource corpora, these methods significantly improve zero-shot and few-shot prediction for typologically diverse and underrepresented languages. The work also provides a comprehensive analysis of how language similarity and pretraining data size affect cross-lingual transfer, offering practical guidelines for robust multilingual NLP.

- *Decomposed Prompting for Multilingual Sequence Labeling:* Extending prompt-based learning to structured prediction tasks, the dissertation proposes the decomposed prompting strategy for sequence labeling. These approaches enable more granular probing of linguistic knowledge in large language models, outperforming iterative baselines in both accuracy and efficiency, and revealing the depth and limitations of cross-lingual generalization in English-centric and multilingual LLMs.

- *Cross-Lingual Knowledge Editing with In-Context Learning:* The dissertation pioneers the study of cross-lingual knowledge editing via in-context learning, introducing BMIKE-53, a comprehensive benchmark covering 53 languages and multiple knowledge editing scenarios. Through extensive experiments, it examines how model scale, demonstration design, and linguistic properties influence the reliability and generalization of knowledge editing across languages.

2. **Prompt-Based Fine-Tuning for Zero-Shot Cross-Lingual Transfer**

- *Prompt-Based Fine-Tuning for Cross-Lingual Transfer:* The dissertation systematically compares prompt-based and vanilla fine-tuning for zero-shot cross-lingual transfer, introducing the PROFIT pipeline and providing empirical evidence that prompt-based fine-tuning consistently outperforms vanilla fine-tuning, especially in few-shot scenarios and for languages with higher similarity to the source language. This work also elucidates the factors, such as language similarity and pretraining data size, that govern transfer effectiveness.

- *Token-Level Prompt Decomposition Fine-Tuning for Structured Prediction:* Building on the above, the dissertation develops ToPro, a token-level prompt decomposition method for cross-lingual structured prediction, such as part-of-speech (POS) tagging and named entity recognition (NER). ToPro achieves state-of-the-art performance on NER and POS tagging, particularly for languages that are typologically distant from English, and provides a robust and interpretable framework for structured prediction in multilingual settings.

- *Cross-Lingual Parsing for Historical Languages:* Parsing historical languages is a challenging structured prediction task. This dissertation presents a delexicalized cross-lingual constituency parser for Middle High German, leveraging modern German resources and linguistic continuity. This approach provides a tool for syntactic analysis in ancient languages, demonstrating the broader applicability of cross-lingual transfer methods beyond contemporary languages.

3. Efficient NLP Methods for Low-Resource Settings

- *Unified Data Augmentation for Multi-Domain Dialogue:* The dissertation introduces AMD^{2}G, a unified data augmentation framework for low-resource multi-domain dialogue generation. By decoupling domain-agnostic and domain-specific features through dedomaining and sequential training, AMD^{2}G achieves superior performance across multiple domains and models, and provides a principled approach to leveraging shared patterns for data-scarce applications.

- *GNN-Based Parameter-Efficient Fine-Tuning:* Inspired by information flow theory, the dissertation proposes GNNavi, a parameter-efficient fine-tuning method that integrates a graph neural network layer into large language models. GNNavi explicitly controls information aggregation in prompts, achieving state-of-the-art performance and training efficiency in few-shot classification tasks, and outperforming existing parameter-efficient methods such as LoRA, Prefix-Tuning, and Adapters.

4. Human-Inspired Analysis and Interpretability of Language Models

- *Human-Inspired Probing Methods:* The dissertation advances the scientific understanding of language models by distinguishing between performance and competence through human-inspired probing paradigms. By using minimal pair probing, it reveals that LLMs encode form more robustly than meaning, and that instruction tuning improves performance but not underlying competence. The work also introduces new multilingual minimal pair datasets for Chinese and German, enabling cross-linguistic analysis of form and meaning representations.

- *Mechanistic Understanding and Editing of Language Confusion:* Addressing a critical failure mode in English-centric LLMs, the dissertation provides the first mechanistic interpretability study of language confusion. Through layer-wise and neuron-level analysis, it identifies critical neurons responsible for unintended language switching and demonstrates that targeted neuron editing can mitigate confusion without harming general competence or fluency. This approach matches the effectiveness of multilingual alignment while preserving cleaner output quality, highlighting the promise of neuron-level interventions for robust and interpretable multilingual NLP.

These contributions are not isolated; rather, they form a coherent and progressive narrative. Foundational advances in prompt-based modeling and calibration enable robust cross-lingual transfer, which is further extended to historical and low-resource languages through scalable adaptation and efficient fine-tuning. Practical frameworks for data augmentation and parameter-efficient adaptation address low-resource deployment challenges, while human-inspired probing and mechanistic analysis deepen our scientific understanding and guide the design of more reliable, interpretable, and inclusive language technologies. Collectively, this dissertation pushes the boundaries of multilingual and low-resource NLP, offering both practical tools and theoretical insights for the next generation of language models.

1.4 Outline of the Dissertation

This dissertation is structured to provide a logical and progressive exploration of efficient and human-inspired NLP methods for multilingual and low-resource settings. The chapters and their contents are organized as follows:

- **Chapter 2: Background and Related Work**
  This chapter reviews the evolution of NLP technologies, with a focus on the development and limitations of large language models (LLMs), multilingual pretrained language models (MPLMs), and current approaches to cross-lingual transfer. It also summarizes foundational work in prompt-based learning and information retrieval for NLP, setting the scientific and practical context for the dissertation's research questions.

- **Chapter 3: Prompt-Based Learning for Multilingual Prediction**
  This chapter investigates how prompt-based learning can be enhanced for multilingual and low-resource settings. It introduces and evaluates probability calibration techniques to mitigate label bias in masked token prediction (§3.1), and proposes cross-lingual retrieval-augmented prompting (PARC) to improve zero-shot performance on low-resource languages (§3.2). The chapter further explores decomposed prompting for structured prediction, revealing the depth of linguistic structure knowledge in LLMs (§3.3), and presents a comprehensive benchmark for in-context cross-lingual knowledge editing (BMIKE-53), analyzing the factors that govern reliable knowledge transfer across languages (§3.4). This chapter focuses on the training-free part of prompt-based learning, i.e., not involving parameter updating, using the in-context learning paradigm instead.

- **Chapter 4: Prompt-Based Fine-Tuning for Zero-Shot Cross-Lingual Transfer**
  This chapter extends the prompt-based approach to fine-tuning, systematically comparing prompt-based and vanilla fine-tuning for cross-lingual transfer across a range of tasks and languages (§4.1). It introduces token-level prompt decomposition (ToPro) for robust sequence labeling in zero-shot settings and demonstrates its effectiveness, particularly for typologically distant languages (§4.2). The chapter also addresses cross-lingual transfer for historical languages, presenting a delexicalized constituency parser for Middle High

German, and showing how cross-lingual transfer techniques can enable syntactic parsing in the absence of annotated treebanks (§4.3).

- **Chapter 5: Efficient NLP Methods for Low-Resource Settings**
  This chapter presents efficient data augmentation and parameter-efficient adaptation methods. It proposes AMD^{2}G, a unified data augmentation framework for low-resource multi-domain dialogue generation, and demonstrates its effectiveness across several domains and models (§5.1). The chapter also introduces GNNavi, a parameter-efficient fine-tuning method inspired by information flow theory, which integrates Graph Neural Networks into LLMs to guide information aggregation in prompt-based learning, achieving superior performance with minimal parameter updates (§5.2).

- **Chapter 6: Human-Inspired Understanding of Language Models**
  This chapter shifts the focus from algorithmic performance to interpretability and human-inspired analysis. It employs neuro- and psycholinguistic paradigms to probe the internal representations of LLMs, distinguishing between linguistic form and meaning, and revealing the gap between performance and true competence (§6.1). The chapter also presents a mechanistic interpretability study of language confusion in English-centric LLMs, identifying the roles of confusion points and proposing targeted neuron-level interventions that robustly mitigate confusion while preserving general competence and output quality (§6.2).

- **Chapter 7: Conclusion**
  The concluding chapter synthesizes the dissertation's main findings and discusses the future work.

Chapter 2

Background and Related Work

This chapter surveys the key developments and current landscape of natural language processing (NLP) relevant to efficient and human-inspired multilingual modeling. We begin by reviewing the evolution of language models, from early statistical and neural models to the rise of transformer-based pre-trained and large language models, and highlighting their transformative impact on NLP applications and their limitations, particularly for less-resourced languages. The chapter then explores the unique challenges and progress in multilingual NLP, including the architecture and training of multilingual pretrained language models, cross-lingual transfer learning paradigms, and the global distribution of language resources in the context of multilingual NLP technologies. We further introduce the emergence of prompt-based learning as a new paradigm for leveraging language models, with a focus on its relevance for multilingual and low-resource scenarios. Finally, we discuss advances in information retrieval techniques that support knowledge-augmented NLP and cross-lingual applications. Together, these topics establish the scientific and practical context for the efficient, robust, and human-centered approaches developed in the remainder of this dissertation.

2.1 Pre-Trained Language Models

Language models (LMs) are foundational to contemporary research in computational linguistics and natural language processing (NLP). By leveraging various types of LMs, a wide range of NLP tasks can be effectively addressed. Each significant advancement in LMs has led to major breakthroughs in the field.

The most recent transformation in LMs is marked by the rise and widespread adoption of transformer-based models trained on massive language corpora, as well as the emergence of large language models (LLMs). Transformer-based LLMs, such as BERT, GPT-2, GPT-3, GPT-4, and Llama 3, have dramatically advanced the capabilities of NLP systems, enabling applications ranging from chatbots and programming assistants to document summarization and translation (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020). These models are characterized by their immense scale, often comprising billions or even hundreds of billions of parameters, and are pretrained on vast and diverse datasets (Liu et al., 2019b; Lan et al., 2020; Sanh et al., 2019; Yang et al., 2019b).

A defining feature of LLMs is their ability to generalize across a wide array of tasks through prompt-based learning, in few-shot or even zero-shot settings, without the need for extensive task-specific fine-tuning. The transformer architecture, with its self-attention mechanism, enables these models to capture complex dependencies and contextual relationships within language, supporting both understanding and generation tasks (Vaswani et al., 2017). Recent research has also focused on extending the context window and improving the efficiency of LLMs, addressing challenges such as long-context comprehension and computational resource demands (Zhao et al., 2023).

While the trend toward ever-larger PLMs and LLMs has driven remarkable progress, it has also raised important questions regarding their risks, including ethical concerns, biases, and the environmental impact of large-scale training. As LLMs become increasingly integrated into real-world applications, ongoing research continues to explore both their potential and their limitations (Bender et al., 2021).

2.1.1 Language Model Development

Language models (LMs) are statistical models that assign probabilities to sequences of words (Jurafsky and Martin, 2000). Their primary function is language prediction, estimating how likely a given sentence is to be correct or natural. Given a sentence S composed of n words, $S = x_1x_2x_3\cdots x_n$, the objective of a language model $\mathcal{L}$ is to compute the probability of the sentence S. By applying the chain rule of probability, this probability can be decomposed into the product of conditional probabilities for each word given its preceding context, as shown in Equation (2.1). However, calculating the probability of a word based on the entire preceding context is computationally complex. To simplify this, the Markov assumption is often used, which limits the context to a fixed number of previous words, such as just the immediately preceding word.

$$\begin{aligned}
P(S) &= P(x_1x_2x_3\cdots x_n) \\
&\overset{*}{=} P(x_1) \times P(x_2|x_1) \times P(x_3|x_1x_2) \times \cdots \times P(x_n|x_1\cdots x_{n-1}) \\
&\overset{**}{=} P(x_1) \times P(x_2|x_1) \times P(x_3|x_2) \times \cdots \times P(x_n|x_{n-1})
\end{aligned} \tag{2.1}$$

* *Applying the chain rule of probability*
** *Applying the Markov assumption*

Over the years, several types of language models have been developed, including n-gram models and neural network-based models.

N-gram LM The n-gram model is a fundamental type of LM, where an n-gram refers to a sequence of n words. N-gram LMs estimate the probability of a word based on its preceding $(n-1)$ words. For example, a bigram model (where $n = 2$) predicts each word based only on the previous word, as shown in Equation (2.1) under the Markov assumption. These probabilities are typically estimated from corpus statistics by counting the frequency of n-grams.

The quality of a language model can be evaluated using both extrinsic and intrinsic metrics. Extrinsic evaluation measures the impact of the LM on downstream tasks, such as machine translation or speech recognition. Intrinsic evaluation, on the other hand, assesses the model independently of any specific application. Perplexity is a widely used intrinsic metric, defined as the normalized inverse probability of the test set, as shown in Equation (2.2).

$$PP(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, \cdots, w_N)}} \tag{2.2}$$

A high-quality LM assigns higher probabilities to correct or natural sentences, resulting in lower perplexity scores. Thus, lower perplexity indicates a better understanding of language.

2.1.2 Word Embeddings and Deep Learning Models

Word embedding technology is fundamental for applying deep learning and neural network methods to language processing. Deep learning operates by processing data in numerical form, where various types of data are input into neural networks and transformed through multiple layers to produce outputs. These outputs can be used for a range of language tasks, such as classification, sequence labeling, and language generation. To enable neural networks to address NLP problems, it is essential to represent language text in a format that can be directly processed by these models. In this context, word representation refers to the digital encoding of words.

One-Hot Encoding A straightforward approach is to assign each word in the vocabulary a unique ID and represent it as a one-hot vector, which contains zeros in all positions except for a single one at the index corresponding to the word's ID. If the vocabulary size is $|V|$, each word vector has length $|V|$. Stacking all word vectors forms a $|V| \times |V|$ diagonal matrix with ones on the diagonal.

While simple, one-hot encoding has two major drawbacks. First, it leads to a parameter explosion, as the dimensionality of each word vector equals the vocabulary size, which can be extremely large. Second, since all vectors are orthogonal, one-hot encoding fails to capture any notion of word similarity. To address these issues, more effective word representations are needed. Ideally, word vectors should have a much lower dimension, $|D|$, typically in the range $50 \leq |D| \leq 1000$, and should encode semantic similarity, which can be measured by the cosine similarity between vectors, as shown in Equation (2.3).

$$\cos(\mathbf{w}^{(i)}, \mathbf{w}^{(j)}) = \frac{\mathbf{w}^{(i)T}\mathbf{w}^{(j)}}{||\mathbf{w}^{(i)}||_2 \cdot ||\mathbf{w}^{(j)}||_2} \tag{2.3}$$

Static Word Embedding Static word embeddings address both the dimensionality and similarity issues. Based on the distributional hypothesis—"a word is characterized by the company it keeps" (Firth, 1957)—these methods learn dense vector representations for words. The Word2vec model (Mikolov et al., 2013a) is a classic example, trained using negative sampling. In Word2vec, word vectors are model parameters learned from context information. Two main training objectives are used: the skip-gram model, which predicts context words given a target word, and the Continuous-Bag-Of-Words (CBOW) model, which predicts the target word from its context. FastText (Bojanowski et al., 2017) extends Word2vec by representing words as bags of character n-grams, allowing it to generate embeddings for out-of-vocabulary words by composing them from subword vectors. GloVe (Pennington et al., 2014) further improves on Word2vec by incorporating both local and global corpus statistics, using a word co-occurrence matrix to capture semantic relationships.

Contextualized Word Embedding Unlike static embeddings, contextualized word embeddings assign each token a representation that depends on the entire input sentence. These embeddings are derived from language models. ELMo (Peters et al., 2018) is a notable example, producing embeddings from a bidirectional language model that combines forward and backward LMs with a context-independent character-based representation. In transformer-based language models, the embedding layer serves as the first layer, and its parameters are learned from the language model itself. With the embedding layer parameters, contextualized word embeddings are calculated.

Neural Network Based LM Neural network-based language models emerged with the adoption of neural network methods in NLP. These models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997), predict the next word in a sequence based on previous words using neural architectures. Neural network-based LMs utilize word embeddings as semantically meaningful input vectors.

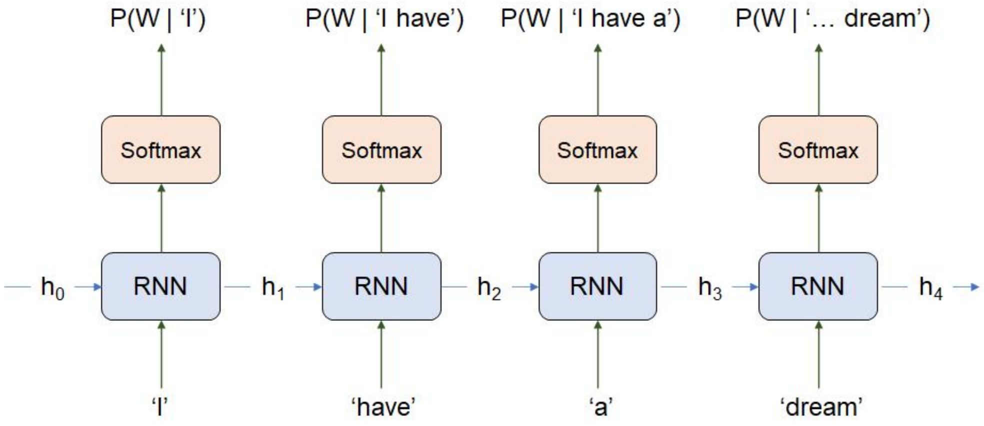


Figure 2.1: An example of RNN Language Model Structure. The input words are first mapped to word IDs and then to word embeddings.

Figure 2.1 illustrates an example of a neural network language model using an RNN structure. The hidden states $h_0, h_1, \cdots$ in the RNN layer transmit information from all previous words to the next state, enabling the model to predict the next word based on the entire preceding context.

However, RNNs face a significant limitation in processing long sequences. At any given time step j, all past input information is compressed into a single hidden state, which becomes a bottleneck for long sequences. The attention mechanism (Bahdanau et al., 2014) was introduced to address this issue, allowing the model to focus on relevant past encoder states and disregard less important information.

2.1.3 Transformer and Language Models

Transformer-Based LM The transformer-based language model is built upon the transformer architecture (Vaswani et al., 2017), which leverages the self-attention mechanism as its core component. Unlike the attention mechanism introduced by Bahdanau et al. (2014), the transformer architecture relies solely on attention mechanisms, without recurrence or convolution. The transformer consists of two main parts: the Encoder and the Decoder, each composed of modules that can be stacked multiple times. The essential components of each module are multi-head attention and feed-forward layers. This structure enables the transformer to provide powerful word representations and serves as the foundation for most pretrained language models.

The paradigm of pretrained language models (PLMs) based on the transformer structure has become dominant in the NLP field in recent years. With advances in computational resources, researchers are able to train deeper and more complex models on increasingly large corpora. Extensive studies have demonstrated that PLMs can capture linguistic features and general knowledge from training data and encode this information into their large-scale parameters.

PLMs are trained using various pretraining tasks. Depending on the pretraining method, PLMs can be categorized as masked language models, left-to-right language models, and encoder-decoder models. Table 2.1 provides an overview of these types and representative examples for each.

Type	Model	Parameters	Dataset Size
Encoder-only	BERT (Devlin et al., 2019)	3.40E+08	16GB
	RoBERTa (Large) (Liu et al., 2019b)	3.55E+08	161GB
	DistilBERT (Sanh et al., 2019)	6.60E+07	16GB
	ALBERT (Lan et al., 2020)	2.23E+08	16GB
Decoder-only	XLNet (Large) (Yang et al., 2019b)	3.40E+08	126GB
	GPT-3 Brown et al. (2020)	1.75E+11	570GB
Encoder-Decoder	BART (Large) (Lewis et al., 2020)	4.00E+08	161GB
	T5-11B (Raffel et al., 2020)	1.10E+10	754GB

Table 2.1: An Overview of Different Types of PLMs.

Encoder-only LM Encoder-only language models are usually masked language models, a kind of auto-encoder models that typically employ a bidirectional objective function, known as masked language modeling (MLM), during pretraining. The MLM objective is to predict masked portions of text based on their surrounding context.

A prominent example of a encoder-only LM is BERT (Devlin et al., 2019), which uses the transformer architecture and learns bidirectional encoder representations. BERT introduces two pretraining objectives: masked language modeling (MLM) and next sentence prediction (NSP). MLM involves predicting randomly masked tokens in a sentence, with 15% of tokens in the training corpus replaced by a mask. NSP requires the model to predict whether a given sentence follows another, with negative samples generated by randomly selecting sentences from the corpus. BERT is pretrained on BooksCorpus and English Wikipedia, and uses the WordPiece tokenization method (Schuster and Nakajima, 2012).

Other notable encoder-only LMs include BERT variants and ERNIE (Zhang et al., 2019b).

Decoder-only LM Decoder-only language models are usually left-to-right language models, a kind of auto-regressive models that predict the next word in a sequence or assign a probability $P(\mathbf{x})$ to a sequence of words $\mathbf{x} = x_1 \cdots x_n$. The likelihood $P(\mathbf{x})$ is factorized using the chain rule in a left-to-right manner:

$$P(\mathbf{x}) = \prod_{t=1}^{T} p(x_t|\mathbf{x}_{<t})$$

Auto-regressive LMs can only predict in one direction, either forward or backward. XLNet (Yang et al., 2019b) is an example of a left-to-right LM, using permutation language modeling as its pretraining objective. In XLNet, the factorization order is permuted rather than the sequence order, allowing the model to achieve bidirectionality while maintaining an auto-regressive objective.

Other modern decoder-only LMs include GPT-3 (Brown et al., 2020) and other GPT-like models.

Encoder-Decoder LM Encoder-decoder models use a language model to compute contextual embeddings for the input tokens x and a decoder language model to generate an output text y conditioned on the input. The decoder accesses the input token embeddings via cross-attention. The encoder and decoder do not share parameters. The T5 model (Raffel et al., 2020), or Text-to-Text Transfer Transformer, is a comprehensive encoder-decoder transformer architecture. T5 reformulates all downstream NLP tasks as text-to-text problems, scaling both dataset and parameter sizes significantly, from BERT-sized models up to over 750GB of data and 11 billion parameters.

Additionally, the encoder-decoder structure is widely adopted in other PLMs such as BART (Lewis et al., 2020), MASS (Song et al., 2019), and their variants.

2.1.4 Large Language Models

Large language models (LLMs) represent a significant advancement in the NLP field, building upon the foundation of PLMs by dramatically increasing both model and data scale. Researchers have observed that scaling up PLMs—whether by increasing the number of parameters or the size of training data—consistently leads to improved performance on a wide range of downstream tasks, a phenomenon described by the scaling law (Kaplan et al., 2020). Notable examples include GPT-3 (Brown et al., 2020), with 175 billion parameters, and PaLM (Anil et al., 2023), with 540 billion parameters. These large-scale models exhibit behaviors and capabilities distinct from their smaller counterparts, such as BERT (330M parameters) or GPT-2 (1.5B parameters), and demonstrate so-called emergent abilities (Wei et al., 2022a). For instance, GPT-3 is capable of few-shot learning through in-context learning, a capability not observed in smaller models like GPT-2.

The term “large language models” (LLMs) has thus been adopted to describe these exceptionally large PLMs, which have attracted substantial research attention in recent years. A prominent application of LLMs is ChatGPT (Ouyang et al., 2022), which adapts the GPT series for dialogue and demonstrates remarkable conversational abilities with humans. The release of ChatGPT has led to a sharp increase in research activity and publications related to LLMs, reflecting their growing impact and importance in the field.

It is important to note that the concept of language modeling is not new, but has evolved significantly alongside advances in artificial intelligence. Early language models were primarily designed to model and generate text data, supporting specific tasks such as information retrieval or language detection. With the advent of neural language models, the focus shifted to learning task-agnostic representations, reducing the need for manual feature engineering. The introduction of PLMs enabled the learning of context-aware representations that could be fine-tuned for various downstream tasks.

The latest generation of LLMs leverages the scaling effect to further enhance model capacity, positioning these models as general-purpose task solvers. This evolution marks a fundamental shift from simple language modeling to complex task solving, greatly expanding both the scope of tasks that language models can address and the level of performance they can achieve. A defining feature of LLMs is their versatility: they can be applied to a wide array of applications, including text generation, translation, summarization, question answering, and dialogue systems. Moreover, LLMs have demonstrated emergent abilities such as reasoning and generalization to unseen tasks, which were not explicitly programmed during training. These capabilities have positioned LLMs as foundational models in artificial intelligence, driving rapid progress and innovation across research and industry.

Despite their impressive achievements, LLMs also present new challenges, including high computational and hardware requirements, increased training and inference costs, ignorance of underrepresented language users, and concerns regarding alignment, safety, and AI fairness and ethics. As research continues, efforts are being made to improve the efficiency, accessibility, and responsible deployment of LLMs, ensuring that their benefits can be widely realized.

2.2 Multilingual Natural Language Processing

2.2.1 Multilingual Pretrained Language Models (MPLMs)

Motivation of MPLMs The advent of pretrained language models (PLMs), trained on vast amounts of unlabeled raw language data, has significantly transformed the research paradigm in NLP in recent years. Through transfer learning, PLMs have achieved remarkable performance across a variety of downstream NLP tasks, even when only limited annotated data is available. However, much of this progress has been concentrated on English text, leaving low- and medium-resource languages with limited benefits from these advancements.

To address this imbalance, multilingual pretrained language models (MPLMs) have been developed. MPLMs are designed to process multiple languages with comparable effectiveness, thereby extending the reach of PLMs beyond high-resource languages. One of the key advantages of MPLMs over monolingual PLMs is the reduction in the number of models that need to be pretrained and maintained. Furthermore, MPLMs possess cross-lingual transfer capabilities, enabling low- and medium-resource languages to benefit in areas such as machine translation, zero-shot task transfer, and typological research.

MPLMs are constructed by extending the principles of PLMs. Instead of relying solely on monolingual training corpora, MPLMs are pretrained on multilingual unlabeled corpora, mapping representations from different languages into a shared semantic vector space. This approach results in multilingual word embeddings that are jointly learned across languages. The architecture and pretraining objectives of MPLMs remain consistent with those of PLMs. However, MPLMs typically require a much larger vocabulary to accommodate multiple languages. For example, the base version of BERT has a vocabulary size of 28,996, whereas the base version of multilingual BERT (mBERT) expands this to 119,547 to support a broader range of languages.

Some Typical MPLMs

mBERT Multilingual BERT (mBERT) (Devlin et al., 2019) is designed to process the 100 largest languages by Wikipedia size, using the corresponding Wikipedia corpora of these languages as its pretraining dataset. Like BERT, mBERT employs masked language modeling (MLM) and next sentence prediction (NSP) as its pretraining tasks, and is trained in a self-supervised manner.

Given the significant variation in Wikipedia sizes across languages, low-resource languages may be underrepresented in the model, while there is a risk of overfitting for languages with very small corpora. To address this, sampling strategies are applied when constructing the multilingual pretraining dataset. Exponential smoothing is used to under-sample high-resource languages (such as English) and over-sample low-resource languages (such as Icelandic). This technique modifies the sampling probability distribution for each language. Initially, the sampling probability matches the frequency of each language in the corpus (e.g., English accounts for 21% of the total corpus and thus has a 21% sampling probability). After exponential smoothing, the frequency of each language is raised to a power S (e.g., $S = 0.7$) and then normalized. For example, if English is originally 1,000 times more likely to be sampled than Icelandic, after smoothing, it becomes only 100 times more likely.

mBERT uses the WordPiece tokenization method (Schuster and Nakajima, 2012), consistent with the original BERT model.

XLM The XLM model (Lample and Conneau, 2019) is another transformer-based MPLM. Similar to BERT, XLM uses MLM as a pretraining objective. In addition, XLM introduces Translation Language Modeling (TLM) as a second pretraining objective to enhance cross-lingual representation learning. For MLM pretraining, XLM uses Wikipedia as the dataset, while for TLM, it utilizes parallel corpora across different languages.

The basic XLM model covers 15 languages, with two extended versions supporting 17 and 100 languages, respectively. The extended versions are pretrained only with the MLM objective, without TLM. XLM adopts Byte Pair Encoding (BPE) (Gage, 1994; Sennrich et al., 2016b) for tokenization, and the vocabulary size of the extended version is approximately 200,000.

XLM-R XLM-R (Conneau et al., 2020) is the multilingual counterpart of RoBERTa (Liu et al., 2019b). Unlike the original XLM, XLM-R does not use TLM as a pretraining objective. Instead, it is pretrained in the same manner as RoBERTa, utilizing 2.5TB of filtered web-crawled data from CommonCrawl, covering 100 languages. XLM-R features a vocabulary size of up to 250,000, compared to the 50,000-word vocabulary of the original RoBERTa.

M2M100 M2M100 (Fan et al., 2021) is a multilingual translation model with an encoder-decoder (seq2seq) architecture, designed for many-to-many multilingual translation. It supports translation between any pair of 100 languages, resulting in $100 \times 99 = 9,900$ possible translation directions. When using M2M100 for translation, the target language ID is used as the first token in the input sequence.

mBART-50 mBART (Liu et al., 2020) is the multilingual version of BART (Lewis et al., 2020), a sequence-to-sequence denoising auto-encoder pretrained on large-scale monolingual corpora in multiple languages using the BART objective. mBART employs a denoising pre-training approach, reconstructing original texts from corrupted inputs. mBART-50 extends this approach to support translation between 50 languages, demonstrating that multilingual translation models can be realized through multilingual fine-tuning. Unlike standard finetuning, which typically focuses on a single translation direction, mBART-50 enables fine-tuning across multiple directions simultaneously.

For multilingual denoising pretraining, all monolingual corpora are concatenated into a single dataset $D = \{D_1, D_2, \cdots, D_n\}$, where D_i is the monolingual corpus for language i. Source texts are corrupted using two noise types: sentence permutation and word-span masking. The objective is to reconstruct the original text. Similar to M2M100, mBART-50 uses a special token at the beginning of the input to indicate the target language.

BLOOM BLOOM (BigScience Large Open-science Open-access Multilingual Language Model) is a multilingual large language model (LLM) developed through the collaborative efforts of the BigScience project (Workshop et al., 2022). BLOOM is a decoder-only autoregressive model with 176 billion parameters, designed to generate and continue text in response to prompts. What distinguishes BLOOM is its open-access and open-science approach: both the model and the code base, as well as the data used for training, are distributed under free licenses, making advanced language modeling technology accessible to a broader research community.

BLOOM is proficient in 59 languages, including a wide range of high-, medium-, and low-resource languages. The model was trained on a massive multilingual dataset, with a focus on inclusivity and diversity in language coverage. This enables BLOOM to excel in multilingual content generation and cross-lingual tasks, supporting research and applications that require robust performance across different languages.

English-Centric and Multilingual LLMs English-centric large language models (LLMs) are predominantly pretrained on extensive English text corpora, though they are also exposed to a limited amount of multilingual data. For instance, LLaMA (Touvron et al., 2023a) is trained on over 1.4 trillion tokens, with less than 4.5% comprising multilingual data from 20 different languages. LLaMA 2 (Touvron et al., 2023b) increases linguistic diversity, covering 27 languages, each contributing more than 0.005% of the total data. Mistral 7B (Jiang et al., 2023), a state-of-the-art English-centric LLM, achieves high performance and efficiency by employing advanced attention mechanisms such as Sliding Window Attention (SWA) (Child et al., 2019), which enables faster inference.

The tokenizers used in English-centric LLMs are often designed to support byte-level encoding (Workshop et al., 2022; Zhang et al., 2022a; Touvron et al., 2023a), allowing these models to process a wide variety of scripts beyond the Latin alphabet. To further enhance multilingual robustness, the Byte-level Byte-Pair-Encoding (BBPE) algorithm (Sennrich et al., 2016b; wan, 2020) is commonly adopted for tokenization. BBPE can decompose UTF-8 characters not present in the model's vocabulary into their constituent bytes, equipping LLMs with the flexibility to handle scripts from any language, even those not encountered during training. Thus, the combination of limited multilingual data exposure and byte-level encoding contributes to the robust multilingual capabilities observed in English-centric LLMs.

LLMs, including multilingual variants, are typically instruction-tuned to improve task understanding and interactivity. Notable examples include BLOOMZ (Muennighoff et al., 2023), derived from BLOOM (Workshop et al., 2022), and mTk (Wang et al., 2022c), based on mT5 (Xue et al., 2021). Instruction tuning is widely used to enhance the performance of LLMs on a variety of tasks (Zhang et al., 2023c). Recent research has further strengthened the multilingual abilities of LLMs through multilingual instruction tuning (Kew et al., 2024; Chen et al., 2023; Shaham et al., 2024). Additionally, multilingual LLMs have been tailored for specific language groups, such as SeaLLMs for Southeast Asian languages (Nguyen et al., 2023).

2.2.2 Multilinguality

Multilinguality refers to the property and capability of multilingual representations, where words from different languages are mapped into a shared vector space and can be directly compared. For instance, in a multilingual model, the German word 'Hund' and the English word 'dog' should be positioned closely in the vector space, reflecting their semantic similarity. MPLMs achieve such multilingual representations through joint pretraining on large-scale multilingual corpora. Recent studies have demonstrated that MPLMs are able to learn high-quality multilingual representations (Lauscher et al., 2020). Remarkably, MPLMs acquire this multilinguality without explicit signals linking different languages during pretraining. Despite the absence of external information about language relationships, MPLMs still exhibit strong multilinguality after training. This phenomenon has prompted extensive research in the NLP community to analyze and explain the origins and mechanisms of multilinguality in MPLMs.

Singh et al. (2019) found that mBERT tends to partition representations by language, rather than forming a unified, shared interlingual space as initially expected. Using projection weighted canonical correlation analysis (PWCCA) (Hotelling, 1992; Morcos et al., 2018), they investigated the relationships between representations of the same data points from different models, in a way that is invariant to affine transformations. Their analysis revealed that mBERT's representations can be partitioned by language, indicating that semantically similar data points are not necessarily closer in a common space. By applying the unweighted pair group method with arithmetic mean (UPGMA) (Sokal, 1958), a hierarchical clustering method, they generated a phylogenetic tree from Layer 6 representations of mBERT, which closely mirrors the linguistic family tree of human languages. At deeper layers, this partitioning becomes more pronounced, suggesting that mBERT abstracts semantic content in a way that reflects natural linguistic differences and similarities. The use of WordPiece tokenization (Schuster and Nakajima, 2012) in BERT, rather than character- or word-level tokenization, is identified as a factor motivating mBERT to uncover these linguistic and evolutionary relationships.

Artetxe et al. (2020) challenged the necessity of joint pretraining and shared vocabulary for mBERT's multilinguality. They proposed an alternative approach by first training a transformer-based masked language model on one language, then transferring it to a new language by learning a new embedding matrix, without shared vocabulary or joint training. Their results were competitive with mBERT, contradicting the hypothesis that shared subword vocabulary and joint training are essential for MPLMs' multilinguality.

Wang et al. (2019b) conducted comprehensive experiments to identify the key components contributing to MPLMs' multilinguality. They examined linguistic properties (such as word-piece overlap, word-order similarity, word-frequency similarity, and structural similarity), model architecture (including model depth, multi-head attention, and number of parameters), and learning objectives (such as NSP, language identity markers, and tokenization types). Their findings indicate that word-piece overlap and multi-head attention are not significant contributors, while structural similarity between languages and model depth are crucial for achieving multilinguality.

Wu et al. (2019d) explored four factors potentially influencing MPLMs' multilinguality: domain similarity, shared vocabulary, shared parameters, and language similarity. Their experiments showed that shared vocabulary and domain similarity are not important, but shared parameters in the top layers are necessary for cross-lingual ability. They further demonstrated that monolingual BERT representations in different languages are isomorphic and can be aligned post-hoc. This suggests that MPLMs leverage universal latent symmetries in embedding spaces and align them automatically during joint training.

Dufter and Schütze (2020) investigated four architectural properties (overparameterization, shared special tokens, shared position embeddings, and random word replacement) and two linguistic properties (word order and corpus comparability) as possible reasons for multilinguality, using a controlled experimental setting. Their results show that limited parameter count, shared special tokens, shared position embeddings, and random masking contribute to multilinguality. Unlike previous studies, they introduced a comprehensive metric, the multilinguality score, to directly measure the model's multilinguality, rather than relying on extrinsic task-based metrics.

Deshpande et al. (2022) focused on the influence of linguistic properties on multilinguality through large-scale experiments. Contrary to earlier findings, they observed that subword overlap significantly affects multilinguality when languages differ in word order. Additionally, they found a strong correlation between word embedding alignment across languages and the degree of multilinguality.

Curse of Multilinguality Conneau et al. (2020) identified a phenomenon in MPLMs where, for a fixed model capacity, cross-lingual transfer performance improves as more languages are added to pretraining, but only up to a certain point. Beyond this, adding more languages leads to a decline in performance, a phenomenon termed the "curse of multilinguality". This issue can be mitigated by increasing model capacity (Artetxe et al., 2020). However, Dufter and Schütze (2020) noted that excessively large model sizes can negatively impact multilinguality, indicating a trade-off between generalization and the degree of multilinguality in MPLMs.

2.2.3 Cross-Lingual Transfer Learning

Transfer learning investigates how machine learning models can be adapted to data outside their original training distribution (Pan and Yang, 2009), including across different tasks, domains, and languages. The motivation for transfer learning arises from the high cost of linguistic annotation and the diversity of NLP tasks. In particular, data scarcity in low-resource languages highlights the need for effective cross-lingual transfer methods. Cross-lingual transfer learning refers to strategies that leverage abundant labeled data from high-resource languages to perform NLP tasks in low-resource languages. In the zero-shot scenario, i.e. **zero-shot cross-lingual transfer learning**, no annotated data from the target language is available, while in the few-shot scenario, i.e. **few-shot cross-lingual transfer learning**, a small amount of labeled data for the target language can be used.

Empirical Study of Cross-Lingual Transfer Cross-lingual word embeddings have been utilized for cross-lingual transfer (Ruder et al., 2019). More recently, pretrained multilingual text encoders have become the standard paradigm for cross-lingual transfer learning. Numerous empirical studies have examined cross-lingual transfer with MPLMs in recent years.

Pires et al. (2019) conducted probing experiments on named entity recognition (NER) and part-of-speech tagging (POS) tasks to evaluate cross-lingual transfer. Their findings show that mBERT enables effective cross-lingual transfer for NER and POS between languages with different scripts and zero lexical overlap, with even better transfer observed for typologically similar languages. Wu and Dredze (2019) extended the investigation to a broader range of tasks, including text classification, dependency parsing, and natural language inference (NLI), covering 39 languages.

With the growing interest in cross-lingual transfer, Hu et al. (2020b) introduced XTREME, a benchmark for evaluating cross-lingual transfer performance with MPLMs. XTREME comprises 9 tasks spanning a subset of 40 languages, categorized into four types: (1) sentence classification, including cross-lingual natural language inference (Conneau et al., 2018) and cross-lingual paraphrase adversaries (Yang et al., 2019a); (2) structured prediction, including POS tagging and NER; (3) sentence retrieval, including parallel sentence extraction and nearest sentence retrieval; and (4) question answering, including cross-lingual question answering (Artetxe and Schwenk, 2019), multilingual question answering (Lewis et al., 2019), and typologically diverse question answering (Clark et al., 2020).

Limitations and Improvement of Cross-Lingual Transfer Despite the impressive performance of zero-shot cross-lingual transfer with MPLMs on many tasks, several limitations have been identified by the NLP community.

Wu and Dredze (2020) compared mBERT's performance between low- and high-resource languages and found that mBERT performs significantly worse on low-resource languages. Lauscher et al. (2020) further demonstrated that both the size of a language's pretraining corpus and the linguistic similarity between source and target languages influence transfer performance. To address these challenges, several approaches have been proposed to improve cross-lingual performance for low-resource target languages.

Pfeiffer et al. (2020b) showed that continued pretraining on monolingual text in the target language using a masked language modeling (MLM) objective can effectively adapt MPLMs to the target language (Howard and Ruder, 2018). Another strategy involves expanding labeled data for low-resource languages by employing machine translation systems. By translating labeled data from the source language into the target language, the pretrained MPLM can be fine-tuned on both source and target language data (Jundi and Lapesa, 2022). Lauscher et al. (2020) demonstrated that leveraging even inexpensive labeled data in low-resource languages yields substantial improvements, suggesting that efforts should be made to move beyond strict zero-shot conditions. In a recent study, Wang et al. (2022b) proposed expanding MPLMs to more low-resource languages through the use of bilingual lexicons annotated and documented by linguists.

2.2.4 Language Resource Distribution

The remarkable success of modern NLP methods, which rely on large-scale labeled and unlabeled corpora, has primarily benefited a small subset of the world's more than 7,000 languages—those with abundant digital resources. The vast majority of languages lack sufficient digital resources and, as a result, have not fully benefited from recent advances in NLP. While the development of MPLMs has partially alleviated this issue, it remains far from resolved. MPLMs trained on up to a hundred languages have demonstrated impressive cross-lingual transfer performance on certain NLP tasks, even in the absence of explicit cross-lingual signals (Wu and Dredze, 2019). However, this strong performance is largely limited to languages with relatively high resources that are included in models like mBERT. Wu and Dredze (2020) found that not all languages are equally represented in MPLMs; when monolingual corpora are small, MPLMs fail to learn high-quality representations for those languages. Furthermore, the pretraining of MPLMs is highly dependent on the availability of monolingual corpora, meaning that most of the world's languages remain uncovered by these models.

This subsection provides an overview of the global distribution of language resources and

Language	ISO	Family	Size Range (GB)
English	en	Indo-European	[11.314, 22.627]
Russian	ru	Indo-European	[2.828, 5.657]
French	fr	Indo-European	
Spanish	es	Indo-European	
German	de	Indo-European	
Chinese	zh	Sino-Tibetan	[1.414, 2.828]
Portuguese	pt	Indo-European	
Polish	pl	Indo-European	
Japanese	ja	Altaic	
Italian	it	Indo-European	
Cebuano	ceb	Austronesian	
Ukrainian	uk	Indo-European	[0.707, 1.414]
Swedish	sv	Indo-European	
Dutch	nl	Indo-European	
Hungarian	hu	Uralic	
Czech	cs	Indo-European	
Catalan	ca	Indo-European	
Arabic	ar	Afro-Asiatic	
Vietnamese	vi	Austroasiatic	[0.354, 0.707]
Turkish	tr	Altaic	
Serbian	sr	Indo-European	
Romanian	ro	Indo-European	
Norwegian	no	Indo-European	
Korean	ko	Altaic	
Indonesian	id	Austronesian	
Hebrew	he	Afro-Asiatic	
Finnish	fi	Uralic	
Persian	fa	Indo-European	
Waray Waray	war	Austronesian	[0.177, 0.354]
Thai	th	Tai-Kadai	
Slovenian	sl	Indo-European	
Slovak	sk	Indo-European	
Serbo Croatian	sh	Indo-European	
Malay	ms	Austronesian	
Armenian	hy	Indo-European	
Croatian	hr	Indo-European	
Galician	gl	Indo-European	
Estonian	et	Uralic	
Greek	el	Indo-European	
Danish	da	Indo-European	
Bulgarian	bg	Indo-European	
Belarusian	be	Indo-European	
Asturian	ast	Indo-European	
Urdu	ur	Indo-European	[0.088, 0.177]
Telugu	te	Dravidian	
Tamil	ta	Dravidian	
Norwegian Nynorsk	nn	Indo-European	
Malayalam	ml	Dravidian	
Mecedonian	mk	Indo-European	
Latvian	lv	Indo-European	
Lituanian	lt	Indo-European	[0.088, 0.177]
Kazakh	kk	Altaic	
Georgian	ka	Caucasian	
Hindi	hi	Indo-European	
Basque	eu	Language Isolate	
Bosnian	bs	Indo-European	
Bengali	bn	Indo-European	
Azerbaijani	az	Altaic	
Uzbek	uz	Altaic	[0.044, 0.088]
Tatar	tt	Altaic	
Tagalog	tl	Austronesian	
Albanian	sq	Indo-European	
Scots	sco	Indo-European	
Occitan	oc	Indo-European	
Marathi	mr	Indo-European	
Latin	la	Indo-European	
Kannada	kn	Dravidian	
Welsh	cy	Indo-European	
Bashkir	ba	Altaic	
Afrikaans	af	Indo-European	
Tajik	tg	Indo-European	[0.022, 0.044]
Swahili	sw	Niger-Congo	
Western Punjabi	pnb	Indo-European	
Punjabi	pa	Indo-European	
Nepali	ne	Indo-European	
Low Saxon	nds	Indo-European	
Burmese	my	Sino-Tibetan	
Mongolian	mn	Altaic	
Lombard	lb	Indo-European	
Kirghiz	ky	Altaic	
Javanese	jv	Austronesian	
Icelandic	is	Indo-European	
Gujarati	gu	Indo-European	
Irish	ga	Indo-European	
West Frisian	fy	Indo-European	
Chechen	ce	Caucasian	
Breton	br	Indo-European	
Bavarian	bar	Indo-European	
Aragonese	an	Indo-European	
Volapük	vo	Artificial	[0.011, 0.022]
Sudanese	su	Afro-Asiatic	
Minangkabau	min	Austronesian	
Malagasy	mg	Austronesian	
Luxembourgish	lmo	Indo-European	
Chuvash	cv	Altaic	
Yoruba	yo	Niger-Congo	[0.006, 0.011]
Sicilian	scn	Indo-European	
Pietmontese	pms	Indo-European	
Ido	io	Artificial	

Table 2.2: List of the 99 Languages with the largest Wikipedia size and the language family they belong to.

examines how this distribution impacts NLP research.

Resource Typology of World Languages The number of languages worldwide is dynamic, with some languages disappearing and new ones emerging. Currently, there are over 7,000 recognized languages. However, the distribution of digital resources is highly uneven. The extent of available resources largely determines how much a language can benefit from modern data-driven NLP methods.

MPLMs such as mBERT cover approximately 100 languages, representing approximately 1% of all languages. These models require large volumes of unlabeled text for pretraining. Yet, widely used resources like Wikipedia and CommonCrawl provide data for only 341[1] and 160[2]

[1] https://en.wikipedia.org/wiki/List_of_Wikipedias

[2] https://commoncrawl.github.io/cc-crawl-statistics/plots/languages

languages, respectively—just about 4% of the world's languages. The Bible, available in 1,600 languages, constitutes the largest parallel corpus in terms of language variety, covering 23% of languages. Nevertheless, more than 70% of languages lack any digital unlabeled data. Despite this, linguistic documentation efforts have produced bilingual lexicons or word lists for about 70%[3] of languages.

Joshi et al. (2020) classified the world's 7,000 languages into six categories based on their digital status and data richness, considering both the quantity of unlabeled and labeled resources:

- **Left-behinds**: Languages with virtually no unlabeled data, largely ignored by language technologies.
- **Scraping-bys**: Languages with some unlabeled data but insufficient labeled data.
- **Hopefuls**: Languages with small labeled datasets, struggling to maintain digital presence.
- **Rising stars**: Languages with strong internet presence and ample unlabeled data, but lacking labeled data for further research.
- **Underdogs**: Languages with large amounts of unlabeled data, comparable to the top group, but limited labeled data.
- **Winners**: Languages with dominant online presence and extensive investment in resources and technology, benefiting most from state-of-the-art methods.

Language Coverage in MPLMs Current multilingual NLP study focuses on the languages included in MPLMs, which can be broadly categorized as low-, medium-, and high-resource languages.

Low-resource languages have attracted significant attention in recent NLP research. Singh (2008) described low-resource languages as resource-scarce, less studied, less computerized, and less privileged. Tsvetkov (2017) defined them as lacking sufficient monolingual or parallel corpora and/or manually crafted linguistic resources for statistical NLP applications. Agić et al. (2016) further characterized truly low-resource languages as those without supporting tools or resources for basic NLP tasks such as segmentation, POS tagging, or dependency parsing. In practice, the classification of low-, medium-, and high-resource languages is based on their representation in the pretraining corpora of MPLMs.

Table 2.2 presents the 99 languages with the largest Wikipedia sizes, along with their language families, as used in the pretraining of mBERT (Devlin et al., 2019). The table lists languages in order of Wikipedia size, with columns for the language name, ISO code, language family, and Wikipedia size range.

Language family reflects the genetic relationships and distances between languages, which is relevant in multilingual NLP, as structural and linguistic similarities can influence cross-lingual performance (Lauscher et al., 2020). Among the languages covered by mBERT, approximately 60% belong to the Indo-European family, with others from Altaic, Sino-Tibetan, Austronesian,

[3] `https://vocab.panlex.org/`

Uralic, Dravidian, Afro-Asiatic, Niger-Congo, and Caucasian families. Notably, two artificial languages—Volapük and Ido—are also included in mBERT.

2.3 Prompt-Based Learning

Prompt-based learning has emerged as a transformative paradigm in NLP, following the development of increasingly large-scale PLMs. It is often regarded as the second major shift in NLP after the pretraining-finetuning paradigm (Liu et al., 2023a). Unlike traditional supervised learning, where a model is trained to predict an output **y** given an input **x** as $P(\mathbf{y}|\mathbf{x})$, prompt-based learning leverages language models to directly predict the probability of text for various NLP tasks. To enable this, the input **x** is reformulated into a cloze-style or text-to-text prompt, allowing the PLM to generate the desired output.

For example, in sentiment analysis, the original input "*This product is amazing.*" is transformed using a template defined by the prompting function $f_{prompt}(\mathbf{x})$ into a new prompt **x'**, as shown in Equation (2.4). In this case, the input becomes "*This product is amazing. In summary, it is a* $[Z]$ *product.*" The label for this example is "1" (positive). Using a verbalizer, which maps labels to words (Schick and Schütze, 2021a), as shown in Equation (2.5), the label "1" is converted to the word "great" and inserted into the prompt. Prompts can also be enriched with additional information, such as task descriptions (Radford et al., 2019) or few-shot examples (Brown et al., 2020).

$$\mathbf{x'} = f_{prompt}(\mathbf{x}) \tag{2.4}$$

$$\mathbf{z} = v(\mathbf{y}) \tag{2.5}$$

2.3.1 Human-Inspired Prompt Learning Development

The evolution of prompt learning is deeply rooted in human-inspired approaches to communication and instruction, which have significantly influenced the development of large-scale PLMs. Early work by Radford et al. (2019) demonstrates that providing explicit task descriptions in natural language prompts enables GPT-2 to perform zero-shot task transfer, mirroring the way humans convey instructions or context to guide understanding. Building on this, Brown et al. (2020) show that GPT-3 could perform a wide range of NLP tasks from just a few examples, a process termed "in-context learning". In this paradigm, the model does not require parameter updates for prediction; instead, it learns from the prompt itself, much like humans learn by observing examples or receiving instructions in context.

GPT-3's ability to achieve strong performance on diverse NLP tasks, including complex reasoning and generation, through in-context learning, highlights the profound impact of human-inspired prompting strategies. However, the immense scale of GPT-3, with its 175 billion parameters, poses practical limitations for widespread adoption. Recognizing these challenges, Schick and Schütze (2021a) demonstrate that prompt-based approaches can be effectively applied to smaller models, such as RoBERTa and ALBERT, by reformulating input examples as cloze-style phrases. This reformulation enables PLMs to better understand and perform specific tasks, drawing inspiration from the way humans fill in missing information based on context. The PET method introduced by Schick and Schütze (2021a) combines prompting with gradient-based optimization, showing that smaller, more efficient models can achieve performance comparable to GPT-3 and even surpass it on benchmarks like SuperGlue with only 32 training examples (Schick and Schütze, 2021c). Beyond classification, Schick and Schütze (2021b) extend these combined methods to generation tasks, such as text summarization and headline generation.

Overall, the development of prompt learning is a testament to the influence of human cognitive strategies, such as learning from instructions, examples, and context, on the design and success of modern NLP models.

2.3.2 Large Language Models and Prompt Engineering

Prompt engineering has become a central technique in leveraging the capabilities of large language models (LLMs), as the way prompts are designed directly influences the performance and reliability of these models. Discrete prompting, also known as hard prompting, involves using natural language templates to describe NLP tasks. In models such as GPT-3 and methods like PET, these prompts are typically human-crafted. Manual template engineering, while effective, can be labor-intensive and may not always yield optimal prompts for every task.

To address these challenges, several approaches have been developed to automate the prompt design process. Gao et al. (2021) utilize the seq2seq pretrained model T5 (Raffel et al., 2020) to search for and generate prompts automatically. Shin et al. (2020) propose using downstream training samples to automatically identify template tokens, while Jiang et al. (2020) employ data mining techniques to discover templates from large text corpora. Paraphrase-based methods further enhance prompt diversity by generating multiple candidate prompts from a single seed prompt (Yuan et al., 2021; Haviv et al., 2021; Zhong et al., 2021).

Beyond discrete prompting, prompt engineering has evolved to include continuous, or soft, prompting. In this approach, prompts are learned directly in the embedding space of the model using stochastic gradient descent (SGD). Continuous prompts can elicit more nuanced knowledge from PLMs compared to discrete prompts (Qin and Eisner, 2021). Techniques such as prefix-tuning, which freezes the parameters of the PLM and only optimizes the prompt embeddings, have demonstrated high parameter efficiency for generation tasks (Li and Liang, 2021).

Integrating the concept of prompting with techniques from other fields has the potential to further enhance performance. Liu et al. (2022e) incorporate unlabeled data into prompt-based learning, aiming to leverage large volumes of unlabeled data to improve the zero-shot capabilities of PLMs without updating model parameters. Their approach combines retrieval-based methods with prompting, opening new avenues for research in prompt-based learning. Inspired by prompt engineering, Wang et al. (2022a) modify the traditional supervised learning process by retrieving similar information from the labeled training set for each input and concatenating it with the retrieved content.

Despite the remarkable advancements in zero- and few-shot learning achieved through prompt-based methods, there remain ongoing discussions and skepticism regarding the underlying mechanisms of prompting. Some studies investigate how prompting enhances PLM performance. For example, Webson and Pavlick (2022) argue that prompt-based models may not truly understand the meaning of prompts, as demonstrated by experiments with misleading and irrelevant prompts. Other researchers suggest that incorporating explanations into prompts can further improve prompt-based learning outcomes.

2.3.3 Advancements in Prompt-Based Learning

Retrieval Augmented Prompt Brown et al. (2020) demonstrate that large-scale pretrained language models like GPT-3 can perform tasks by including input-output examples as context within the prompt. This in-context learning approach concatenates the input with examples randomly selected from the training set. Building on this, recent studies (Gao et al., 2021; Liu et al., 2022a,e) enhance prompts for pretrained models by retrieving semantically similar examples, rather than random ones. These retrieval-augmented methods are applied to discrete prompts, where prompts are represented by tokens instead of continuous vectors. Such retrieval-augmented prompts can be used for both fine-tuning in few-shot settings and for zero-shot learning. Chowdhury et al. (2022) extend this idea by employing a kNN-based retrieval approach to tune soft prompts in a continuous space within a standard supervised training framework.

Multilingual Prompt Learning Despite the notable success of prompting in English, its application in multilingual tasks has not been extensively explored. Research on prompt learning with multilingual pretrained language models (MPLMs) for cross-lingual transfer and low-resource languages remains limited. Zhao and Schütze (2021) applied both discrete and soft prompting techniques to the XNLI task using MPLMs, demonstrating that prompting outperforms finetuning in few-shot cross-lingual transfer and in-language training for multilingual natural language inference. Similarly, Huang et al. (2022) introduced a unified prompt approach for all languages in zero-shot cross-lingual settings with MPLMs. Winata et al. (2021) highlighted the multilingual capabilities of language models primarily trained on English data by providing a few English examples as context and evaluating on non-English data. Recent studies have begun to investigate prompt learning with MPLMs (Zhao and Schütze, 2021; Huang et al., 2022). Subsequent research introduced enhancements such as mask token augmentation (Zhou et al., 2023) and unified multilingual prompts (Huang et al., 2022) for zero-shot cross-lingual transfer.

While these methods have attracted increasing attention, particularly in few-shot scenarios across various NLP tasks, comprehensive investigations into the variations of prompt-based learning methods across different few-shot and full-data settings are still lacking. For example, Tu et al. (2022) proposed an alternative prompting approach for cross-lingual transfer in full-data scenarios, introducing additional prompt parameters to PLMs and updating only these parameters during fine-tuning. More recently, Shi and Lipani (2023) combined prompt-based fine-tuning with continued pretraining, though their work was limited to monolingual settings.

Brown et al. (2020) demonstrated that LLMs like GPT-3 can acquire in-context learning (ICL) abilities for task solving. The advent of multilingual LLMs (MLLMs) has enabled zero-shot cross-lingual ICL, as evidenced by recent benchmarks such as MEGA (Ahuja et al., 2023) and BUFFET (Asai et al., 2024). However, current ICL methods that use text-to-text prompting with fixed output templates for sequence labeling tasks have been shown to “consistently exhibit extremely poor performance" (Asai et al., 2024) when applied to MLLMs, failing to fully exploit their cross-lingual transfer capabilities.

Prompting for Sequence Labeling Prompting methods have rarely been applied to sequence labeling tasks, as most prior work has focused on sentence-level classification. However, several studies have explored prompt-based fine-tuning for sequence labeling. Cui et al. (2021) introduced template-based prompting techniques with the BART model (Lewis et al., 2020) for Named Entity Recognition (NER), employing a rank-based approach that generates a sentence for each possible label and computes the probabilities for prediction, though this can be computationally expensive. Ma et al. (2022) proposed a template-free prompting strategy for few-shot NER, termed entity-oriented language model fine-tuning. Similarly, Ma et al. (2024) utilized a decomposition-based prompting method to fine-tune multilingual encoder models for cross-lingual sequence labeling.

Despite these advances, prompting large language models (LLMs) for sequence labeling remains challenging (Ahuja et al., 2023). While text-to-text prompting is widely used in benchmarking LLMs (Lai et al., 2023a), its application to sequence labeling is hindered by the complexity of output templates, which may not fully capture the capabilities of LLMs (Asai et al., 2024). Iterative structured prompting, specifically designed for sequence labeling, has been introduced to address this issue (Blevins et al., 2023). In this approach, the model predicts the label for each word in the sequence step by step, feeding the predicted label and the next word back into the model for subsequent predictions. However, this token-by-token dependency significantly slows down inference.

Recent work has adapted structural prompting methods for multilingual benchmarking of LLMs (Ahuja et al., 2023). While prompt-based methods have shown promise for sentence-level tasks, their application to token-level sequence labeling remains less explored and presents unique challenges in both efficiency and effectiveness.

2.4 Information Retrieval for Natural Language Processing

In NLP research, information retrieval (IR) methods are frequently employed to gather external knowledge and resources for solving a variety of tasks. This is especially relevant in the era of large language models (LLMs), where neuro-symbolic approaches and retrieval-augmented generation (RAG) benefit from integrating external knowledge sources. In this dissertation, IR methods are also utilized, with a particular focus on cross-lingual retrieval. Retrieval-based approaches can be broadly categorized based on the type of representation used by the retriever: sparse representations, typically based on the bag-of-words (BOW) model (Chen et al., 2017), and dense representations, derived from neural network encoders (Karpukhin et al., 2020).

2.4.1 Sparse and Dense Retrieval Methods

Retrieval with Sparse Representations Sparse representation methods are grounded in the BOW model and are widely applied to large-scale search and open-domain question answering (Chen et al., 2017). In these methods, both queries and documents are represented as high-dimensional, sparse vectors, with weights reflecting term importance. Document ranking is performed using rule-based scoring functions.

For full-text search collections, it is essential to consider term frequency (tf) and document length. The BM25 algorithm (Manning et al., 2008) is a probabilistic model that has been extensively and successfully used across various search tasks. BM25 builds upon the Binary Independence Model (BIM) (Yu and Salton, 1976), incorporating tf and length normalization into its scoring. The BIM score for a document d is based on the inverse document frequency (idf) of the query terms present in the document:

$$RSV_d = \sum_{t \in q \cap d} \log \frac{N}{df_t} \tag{2.6}$$

BM25 refines the idf term $\frac{N}{df}$ in Equation (2.6) by integrating term frequency and document length, as shown in Equation (2.7):

$$RSV_d = \sum_{t \in q \cap d} \log \frac{N}{df_t} \cdot \frac{(k_1 + 1) tf_{td}}{k_1((1-b) + b \times (L_d / L_{ave})) + tf_{td}} \tag{2.7}$$

- tf_{td}: term frequency of term t in document d
- L_d (L_{ave}): length of document d (average document length in the collection)
- k_1: parameter controlling the scaling of term frequency
- b: parameter controlling the scaling by document length

Sparse representations offer strong generalization and efficiency, making them well-suited for searching large-scale document collections.

Retrieval with Dense Representations Dense representations, in contrast, are obtained through latent semantic encoding, typically using neural network encoders pretrained on task-specific data. With the rise of transformer-based models, dense retrieval methods have become a major area of exploration, often complementing traditional sparse retrieval approaches. Dense representations are particularly effective for matching synonyms or paraphrases that do not share common tokens, a scenario where sparse methods often fail. In dense retrieval, such semantically similar terms are mapped to nearby points in the vector space, increasing the likelihood of successful retrieval. As a result, dense retrieval methods often achieve higher recall than sparse methods on tasks such as open-domain question answering (Karpukhin et al., 2020).

However, dense retrieval also faces two main limitations. First, learning high-quality dense vector representations requires large amounts of labeled question-context pairs, necessitating pair data for model training on specific tasks. Second, due to the architectural constraints of transformers, dense retrievers are unable to process very long documents, as the maximum sequence length is limited by the model's configuration.

2.4.2 Sentence Transformers for Retrieval

Words can be represented as word embeddings and directly applied to various NLP tasks or used as inputs for language models. Similarly, mapping sentences or short text passages into a dense vector space, where semantically similar sentences are positioned closely, has broad applications.

Sentence Representation A straightforward approach to obtaining sentence embeddings is to aggregate word embeddings, for example, by averaging the embeddings of all tokens in a sentence. These word embeddings can be static, such as GloVe (Pennington et al., 2014), or contextual, derived from pretrained language models (PLMs). A common method is to input single sentences into BERT and then obtain a fixed-size vector either by averaging the output layer embeddings or by using the embedding of the special $[CLS]$ token. For instance, May et al. (2019) utilize this approach to measure social biases, while Zhang et al. (2020b) sum token similarities to evaluate the similarity between candidate and reference sentences.

Beyond these methods, other strategies have been proposed for sentence representation. Kiros et al. (2015) train an encoder-decoder model to reconstruct surrounding sentences, mapping semantically and syntactically similar sentences to nearby vectors. Conneau et al. (2017) leverage supervised natural language inference datasets to train universal sentence representations using a siamese BiLSTM network with max-pooling. Subsequently, Cer et al. (2018) introduce the Universal Sentence Encoder, a transformer-based model. More recently, sentence transformers have been developed by modifying transformer-based PLMs. Reimers and Gurevych (2019) apply siamese and triplet network structures to BERT and RoBERTa, producing semantically meaningful sentence embeddings that can be compared using cosine similarity. This approach achieves state-of-the-art results on semantic textual similarity (STS) and other transfer learning tasks, and has been extended to various PLMs, resulting in a range of sentence transformer models.

Multilingual Sentence Embeddings As in other areas of NLP, the distribution of language resources is imbalanced for sentence embeddings, with most existing models being monolingual and typically focused on English. To address this, multilingual sentence embeddings have been developed using transfer learning or knowledge distillation techniques, enabling broader language coverage.

Chidambaram et al. (2018) train the Multilingual Universal Sentence Encoder (mUSE) in a multi-task setting, utilizing the SNLI dataset (Bowman et al., 2015) and a large web-crawled question-answering pairs dataset. Reimers and Gurevych (2020) employ multilingual knowledge distillation to train multilingual sentence transformers. In this approach, the MPLM XLM-R serves as the student model $\hat{M}$, while sentence BERT acts as the teacher model M. Training requires parallel sentences in the source and target languages, $((s_1, t_1), \ldots, (s_n, t_n))$, where t_i is the translation of s_i. The student model is trained so that $\hat{M}(s_i) \approx M(s_i)$ and $\hat{M}(t_i) \approx M(s_i)$, using mean squared error (MSE) as the loss function. This method has proven effective for over 50 languages from diverse language families and can be readily extended to additional languages.

2.5 Summary

This chapter has provided a comprehensive overview of the foundational concepts, methodologies, and recent advances that underpin this dissertation's investigation into efficient and human-inspired multilingual NLP for low-resource settings. We began by tracing the evolution of language models, from early statistical n-gram models and static word embeddings to the rise of deep learning, transformer architectures, and large-scale pre-trained and large language models (LLMs). The discussion highlighted how these advances have enabled remarkable progress in NLP, while also introducing new challenges related to model scalability, computational cost, and equitable language coverage.

We then examined the development and impact of multilingual pre-trained language models (MPLMs), detailing their architectures, pretraining strategies, and cross-lingual transfer capabilities. Special attention was given to the limitations posed by the uneven distribution of digital language resources and the "curse of multilinguality", which restricts the benefits of recent models to a small subset of the world's languages. The chapter further reviewed empirical findings on cross-lingual transfer learning, outlined the current typology of language resources, and discussed how linguistic and architectural factors influence model multilinguality.

Next, we introduced the paradigm of prompt-based learning, emphasizing its human-inspired roots and its growing importance for efficient, instruction-driven adaptation of language models to new tasks and languages. The chapter covered both discrete and continuous prompt engineering, advancements in retrieval-augmented prompting, and the emerging application of prompt-based methods to multilingual and sequence labeling tasks.

Finally, we discussed the role of information retrieval in NLP, contrasting sparse and dense retrieval methods, and highlighted the development of sentence-level and multilingual embeddings that support cross-lingual information access and retrieval-augmented generation.

Together, these topics establish the scientific context and motivate the methodological innovations and analyses that follow, laying a strong foundation for the dissertation's contributions to robust, scalable, and human-aligned multilingual NLP.

Chapter 3

Prompt-Based Learning for Multilingual Prediction

Summary of This Chapter

As introduced in Section §2.3, prompt-based learning has rapidly emerged as a transformative paradigm in NLP, particularly with the advent of large language models (LLMs). By reformulating downstream tasks as prompt-driven language modeling problems, this approach enables models to leverage their pre-trained knowledge for zero-shot and few-shot learning, often without the need for additional parameter updates. This property is especially valuable for multilingual NLP, where the scarcity of annotated data in many languages, especially low-resource languages, poses a persistent challenge to the development of inclusive and robust language technologies. In the context of multilingual and low-resource NLP, prompt-based learning offers several key advantages. First, it enables efficient adaptation to new languages and tasks by exploiting the generalization capabilities of pre-trained models, thus reducing the reliance on costly data annotation and model fine-tuning. Second, prompt-based methods are inherently flexible, supporting a wide range of tasks, such as classification, sequence labeling, and knowledge editing, across diverse linguistic settings. Third, by leveraging in-context learning, prompt-based approaches can facilitate cross-lingual transfer, making it possible to extend the benefits of state-of-the-art models to languages and domains that are otherwise underrepresented.

This chapter systematically investigates prompt-based learning for multilingual prediction, with a particular focus on training-free, parameter-frozen methods that utilize in-context learning. We explore the effectiveness of prompt-based approaches across several core multilingual tasks, including text classification, sequence labeling, and knowledge editing. The chapter is structured around four main contributions, each addressing a critical aspect of prompt-based multilingual NLP.

We begin by examining a fundamental limitation of prompt-based learning, i.e., bias in mask token prediction. This bias, often arising from the frequency of label words in the pre-training corpus or from prompt design, can significantly degrade zero- and few-shot performance, particularly in case of low-resource and typologically diverse languages. We introduce and evaluate a suite of calibration methods that adjust the predicted probabilities of label words, demonstrating substantial improvements in multilingual prediction accuracy (§3.1).

To further enhance prompt-based learning for low-resource languages, we propose PARC, a novel pipeline that augments prompts with semantically similar examples retrieved from high-resource language corpora. By incorporating cross-lingual retrieval into the prompt context, PARC enables more effective zero-shot transfer and robust performance gains across multiple tasks and language families, even for languages unseen during pre-training (§3.2).

Recognizing the limitations of standard prompt-based approaches for structured prediction, we introduce a decomposed prompting strategy for sequence labeling tasks such as part-of-speech tagging. By generating individual prompts for each token in a sentence, this method allows for more precise probing and evaluation of the linguistic structure knowledge encoded in LLMs, and reveals new insights into their multilingual capabilities (§3.3).

Finally, we extend prompt-based learning to the domain of knowledge editing, presenting BMIKE-53, a comprehensive benchmark for cross-lingual in-context knowledge editing across 53 languages. We systematically evaluate the ability of LLMs to update and transfer factual knowledge across languages using in-context demonstrations, and analyze the factors that influence the reliability and generalization of knowledge editing in multilingual settings (§3.4).

Through these investigations, this chapter demonstrates how prompt-based learning, when carefully calibrated and augmented, can serve as a powerful and flexible tool for advancing multilingual NLP. The methods and analyses presented here not only improve the practical performance of LLMs in low-resource and diverse linguistic contexts, but also lay a methodological foundation for a deeper understanding of the mechanisms underlying cross-lingual generalization and knowledge transfer.

3.1 Calibration of Prompt-Based Learning: Enhancing the Multilingual Understanding of Encoder Models

This section corresponds to the following work:

> **Ercong Nie**, Helmut Schmid, and Hinrich Schuetze. 2023. Unleashing the Multilingual Encoder Potential: Boosting Zero-Shot Performance via Probability Calibration. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 15774–15782, Singapore. Association for Computational Linguistics.

Declaration of Co-Authorship. I conceived the idea of improving the zero-shot performance of multilingual encoders in language understanding tasks by identifying bias issues in regular prompt-based mask token prediction. I designed and conducted all experiments and drafted the initial version of the manuscript. Helmut Schmid and Hinrich Schütze supervised the project, provided feedback, and contributed to revising the paper draft.

Summary of This Section

Pretrained multilingual encoder models can directly perform zero-shot multilingual tasks or linguistic probing by reformulating the input examples into cloze-style prompts. This is accomplished by predicting the probabilities of the label words at the masked token position, without requiring any updates to the model parameters. However, the performance of this pattern is limited by the model's bias toward predicting label words that frequently occurred during the pretraining. These words typically receive high probabilities. To address this issue, we combine the models with various *calibration* techniques which modify the probabilities of label words predicted by the models. We evaluate the effectiveness of these calibration methods on monolingual encoders as well as multilingual encoders. Across a diverse range of tasks, we achieve substantial performance gains through calibration. Furthermore, with only very few training samples, the trained calibration parameters are able to yield additional enhancements.

3.1.1 Bias in Mask Token Prediction

Prompt-based learning (Brown et al., 2020; Liu et al., 2023a) has emerged as an important research area. By reformulating language understanding into the form of cloze prompts (Schick and Schütze, 2021a,c) or prefix prompts (Li and Liang, 2021; Lester et al., 2021), autoencoding language models (LMs) such as BERT and autoregressive LMs such as GPT can perform zero-/few-shot learning. Recent research demonstrates that multilingual encoder models are capable of accomplishing zero-shot cross-lingual learning (Zhao and Schütze, 2021; Huang et al., 2022) and linguistic probing (Shapiro et al., 2021; Hartmann et al., 2021) by using cloze-style prompts. These prompts consist of an input sample, a task-specific context, and a `mask` token. The encoder model applies Masked Language Modeling (MLM) (Devlin et al., 2019) to generate predictions for the `mask` token using a selection of prescribed candidate tokens from the vocabulary. These predictions can be subsequently utilized for label classification or probing purposes. For example, in order to determine the sentiment of the product review "*Worked as stated!*", we can create the close-style question: "*Worked as stated!* `All in all, it was [MASK]`." and ask the model to predict the probabilities of the verbalizers "*good*" (for the class POS) and "*bad*" (for the class NEG) in the masked token position. The class with the more likely verbalizer is chosen.

However, earlier studies indicate that the output of masked token prediction is ***biased*** towards certain label words in the candidate token list (Weissweiler et al., 2022; Nie et al., 2023a). This bias not only affects the predicted class probabilities (Holtzman et al., 2021; Ahuja et al., 2022), but also deteriorates the model's overall performance (Zhao et al., 2021; Lu et al., 2022). According to Weissweiler et al. (2022) and Zhao et al. (2021), label words with higher frequency in the pretraining corpus tend to be predicted with higher probabilities. Besides, the prompt context can also influence the degree of bias present in the masked token prediction.

Figure 3.1 illustrates the impact of the above-mentioned biases on the model predictions. It shows the results of a binary sentiment analysis task with the $\texttt{BERT}_{\texttt{Base}}$ model. In this example, we use $\{\texttt{good}, \texttt{bad}\}$ as label words for classes $\{\textsc{positive}, \textsc{negative}\}$, and "`[X]` . *All in all, it was* `[MASK]`." as a prompt template. By shifting the threshold for predicting POS from 0.5 to

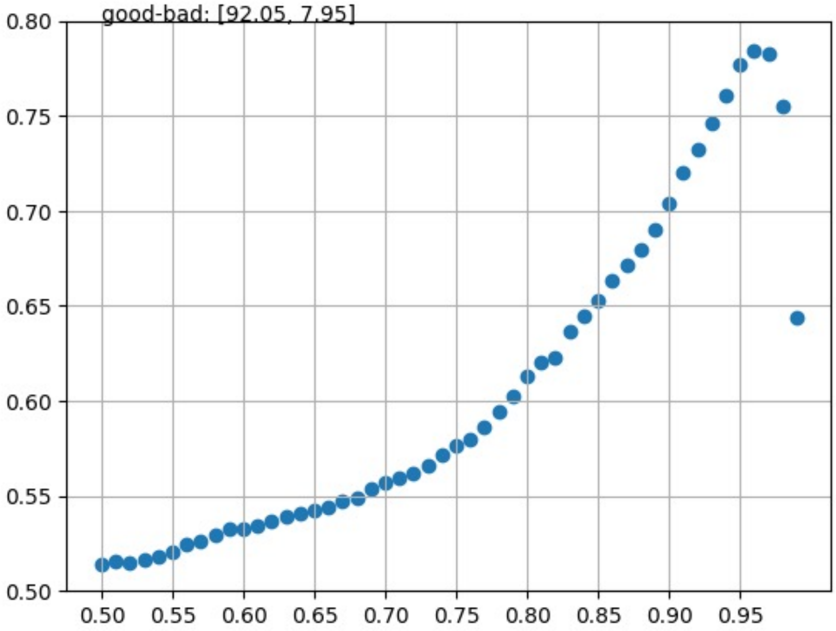


Figure 3.1: Example of the model predictions bias. The graph shows the accuracy on the Amazon polarity test data as a function of the classification threshold. x-axis refers to the threshold probability of `good` to classify examples as POS. The best results are obtained by classifying examples as POS if the probability of `good` exceeds 0.96.

approx. 0.95, the performance can be improved by more than 25%. Given only a `[MASK]` token as input, the model predicts 0.92 and 0.08 as probabilities for `good` and `bad`. To tackle the bias in the distribution of label words, our proposed solution in this work is to combine pretrained encoder models with *calibration* methods.

In this section, we

1. propose a simple yet effective calibration method that involves adding trainable penalties to the probabilities of the label words;

2. demonstrate its effectiveness in achieving performance enhancements comparable to other existing calibration techniques;

3. refine the calibration parameters with only a few training examples for further improvement;

4. boost the zero-shot performance of multilingual encoders by introducing calibration methods.

Method	Probability Calculation	Source
CC	$\tilde{q}(\mathbf{y}\mid x,t) = \mathbf{W}p(\mathbf{y}\mid x,t) + \mathbf{b}$	Zhao et al. (2021)
PMI$_{\text{DC}}$	$\tilde{q}(\mathbf{y}\mid x,t) = log\frac{p(\mathbf{y}\mid x,t)}{p(\mathbf{y}\mid t)}$	Holtzman et al. (2021)
CBM	$\tilde{q}(\mathbf{y}\mid x,t) = \frac{p(\mathbf{y}\mid x,t)}{\frac{1}{\mid X\mid}\sum_{x'\in X} p(\mathbf{y}\mid x',t)}$	Yang et al. (2023b)
Penalty	$\tilde{q}(\mathbf{y}\mid x,t) = p(\mathbf{y}\mid x,t) + \mathbf{p}$	**Proposed by this work**

Table 3.1: Overview of Calibration Methods. **y** refers to the label words. X is the test dataset, x is an input sample, and t is the prompt template.

3.1.2 Calibration Methods

Existing Calibration Methods

Contextual Calibration (CC) Zhao et al. (2021) apply an affine transformation (Platt et al., 1999) to the original probabilities, as the first equation in Table 3.1 shows. The parameters of the affine transformation are obtained from the output probability distribution of the content-free input, e.g., the `mask` token, denoted $\hat{\mathbf{p}}_{cf}$. $\mathbf{W} = \text{diag}(\hat{\mathbf{p}}_{cf})^{-1}$ is the inverse diagonal matrix of $\hat{\mathbf{p}}_{cf}$ and **b** is an all-zero vector.

Domain Conditional Pointwise Mutual Information (PMI$_{\text{DC}}$) Holtzman et al. (2021) adjust the conditional class probability $p(\mathbf{y}|x,t)$ by dividing it with the prior probability $p(\mathbf{y}|t)$ of that class. We estimate $p(\mathbf{y}|t)$ for a given template t using MLM with a prompt created by instantiating the prompt template with an empty input.

Calibration By Marginalization (CBM) Yang et al. (2023b) are inspired by PMI$_{\text{DC}}$. Unlike PMI$_{\text{DC}}$, CBM approximates $p(\mathbf{y}|x,t)$ in a more precise manner by computing its marginalized probability, as the third equation in Table 3.1 shows. For each prediction, the sum probability $\Sigma_{x'\in X}p(\mathbf{y}|x',t)$ are calculated by taking all test inputs into account.

Probability Penalty

Motivated by the observation in Figure 3.1 that a simple shift in the model's output distribution can substantially alleviate the label bias, we propose a penalty-based calibration approach shown in the fourth equation of Table 3.1. The core idea is to introduce penalty terms that is added to each individual label word probability. We initialize the corresponding parameter vector **p** with the negative prior probabilities of the label words. We estimate these prior probabilities using the output distribution of MLM applied to a `[MASK]` token as input.

3.1.3 Experimental Setup

Dataset We first validate the effectiveness of the different calibration methods on several monolingual tasks. We study sentiment analysis using two datasets: binary **Amazon Polarity** (McAuley and Leskovec, 2013) and the English subset of 5-label **Multilingual Amazon Reviews** (Keung et al., 2020), topic categorization using two datasets: the **Ag News** and **Yahoo Answers Topics** (Zhang et al., 2015), sentence pair classification using two datasets: English subsets of **MNLI** Conneau et al. (2018) and **PAWS-X** (Yang et al., 2019a), and 5 datasets from the GLUE benchmark (Wang et al., 2019a): **CoLA** (Warstadt et al., 2019), **MRPC** (Dolan and Brockett, 2005), **QQP**, **RTE** (Dagan et al., 2005), and **WNLI** (Levesque et al., 2012). For the evaluation of multilingual encoders, we use **Multilingual Amazon Reviews**, **XNLI** and **PAWS-X**. Besides, following Nie et al. (2023a), we expand the **AG News** dataset to 25 languages using machine translation to conduct a wide range of cross-lingual analyses.

Setup In our monolingual experiments, we use the pretrained models `bert-base-cased` (Devlin et al., 2019) and `roberta-base` (Liu et al., 2019b). In the multilingual experiments, we use their multilingual counterparts `bert-base-multilingual-cased` and `xlm-roberta-base` (Conneau et al., 2020). We use PyTorch (Paszke et al., 2019) and the HuggingFace framework (Wolf et al., 2020). We repeat each experiment 5 times with different random seeds ({42, 1234, 512, 1213, 421}) and report the mean and variance. To ensure experimental reproducibility, we present the hyperparameter settings used in our study in Table 3.2. We use a batch size of 8 for evaluation. We use a learning rate of 1e-4 for training the calibration parameters. We randomly sample the few-shot training examples from the training sets of each dataset.

Hyperparameter	**Value**
Evaluation batch size	8
Learning rate	1e-4
Random seeds	{42, 421, 512, 1213, 1234}
Maximal sequence length	128
Few-shot numbers	{1, 2, 4, 8, 16}
GPU type	NVIDIA GeForce GTX 1080 Ti
Number of GPU	8

Table 3.2: Overview of hyperparameters.

Prompt Engineering We select a set of prompt templates for the tasks through our preliminary experiments. Table 3.3 shows the prompt templates and the label words used in our experiment.

Baseline To establish a baseline, we initially conduct experiments without employing any calibration methods. Subsequently, we introduce four calibration methods individually and evaluate their impact on the performance. This sequential approach allows us to compare the results and assess the effectiveness of each calibration method in improving the model's performance. Besides, we compare our calibration methods with an NLI-based zero-shot classification baseline proposed by Yin et al. (2019), where they first finetune a pretrained language model on the

Task	Prompt template	Label words
Ag News	`mask` News: `[X]`	‘World’, ‘Sports’, ‘Business’, ‘Tech’
Amazon-P	`[X]`. All in all, it was `mask`.	‘bad’, ‘good’
Amazon-P	`[X]`. All in all, it was `mask`.	‘terrible’, ‘bad’, ‘ok’, ‘good’, ‘great’
XNLI	`[X]`? `mask`, `[Y]`	‘Yes’, ‘Maybe’, ‘No’
Yahoo	`mask` Question: `[X] [Y]`	‘Society’, ‘Science’, ‘Health’, ‘Education’, · · ·
PAWS-X	`[X] . mask[ Y]`	‘Wrong’, ‘Right’
CoLA	`[X]` . It is linguistically`mask`.	‘wrong’, ‘right’
MRPC	`[X]`? `mask`, `[Y]`	‘Wrong’, ‘Right’
QQP	Question 1: `[X]` Question 2: `[Y]` Question 1 and Question 2 are `mask`	‘different’, ‘same’
RTE	`[X]`? `mask`, `[Y]`	‘Wrong’, ‘Right’
WNLI	`[X]`? `mask`, `[Y]`	‘Wrong’, ‘Right’

Table 3.3: Overview of prompt templates.

	Balanced datasets (Acc.)					Imbalanced datasets (F1 Score)						Avg.
	AG News	Amazon-P	Amazon-S	XNLI	Yahoo	Pawsx	CoLA	MRPC	QQP	RTE	WNLI	
BERT$_{\text{Base}}$												
+ *no calib.*	60.2	54.6	24.8	41.3	36.0	31.2	41.2	46.1	26.9	39.5	29.0	39.2
+ *CC*	**74.6**	61.7	27.4	41.4	36.2	31.6	51.1	46.1	26.9	39.5	**43.1**	43.6
+ *PMI$_{DC}$*	62.1	70.8	29.9	37.9	32.1	33.8	**51.3**	44.3	49.5	38.2	30.4	43.7
+ *CBM*	73.6	**71.3**	**33.6**	**42.9**	**45.2**	**49.3**	49.9	**50.6**	**52.6**	**50.9**	42.3	**51.1**
+ *Penalty*	67.9	61.7	26.3	42.6	39.4	31.6	51.1	46.1	26.9	39.5	**43.1**	43.3
RoBERTa$_{\text{Base}}$												
+ *no calib.*	76.2	66.1	24.3	44.0	32.4	31.2	39.6	45.3	26.9	37.1	31.6	41.3
+ *CC*	74.1	**79.5**	20.0	39.8	15.2	33.7	23.6	46.6	39.8	35.9	32.1	40.0
+ *PMI$_{DC}$*	62.3	79.4	**34.2**	45.6	25.3	43.3	43.3	**49.4**	27.1	37.0	30.4	43.4
+ *CBM*	**78.4**	76.5	34.1	**46.4**	**42.9**	**44.4**	**48.2**	47.5	**50.1**	**43.3**	**49.0**	**51.0**
+ *Penalty*	75.6	**79.5**	30.1	41.4	26.9	33.7	23.6	46.6	39.8	35.9	32.1	42.3

Table 3.4: Results of calibration methods on monolingual tasks. Amazon-P refers to Amazon Polarity (binary classification). Amazon-S refers to Amazon Star (5-way classification).

MNLI dataset, then they reformulate common classification tasks to an NLI task format. The input sample is regarded as the premise, while the label serves as the hypothesis. The zero-shot classification is performed by directly comparing the probabilities of predicting `entailment` for all input-label pairs. For this baseline, we fine-tune a `BERT` model and a `RoBERTa` model on the MNLI task. The results for this baseline can be found in Table 3.5.

3.1.4 Results and Analysis

3.1.4.1 Results on Monolingual Encoders

Zero-shot calibration. We first validate the effectiveness of the various calibration methods on monolingual encoders. Table 3.4 shows the results of zero-shot calibration, where we directly calculate the calibrated probabilities without using additional training samples. We report accuracies for evenly distributed datasets and F1 scores for imbalanced datasets. Compared to the uncalibrated baseline systems, we obtain improvements across most of the tasks, except for the *CC* method combined with the `RoBERTa` model. In this specific case, the average performance worsens compared to the *no calibration* baseline due to outlier performance observed in several tasks, such as Yahoo and CoLA.

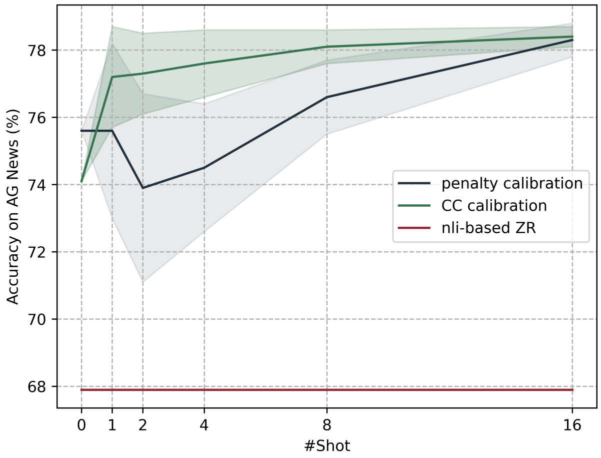


Figure 3.2: Performance and variation of few-shot calibration on the RoBERTa model.

Few-shot samples further boost the performance. As the formulas in Table 3.1 show, *PMI*$_{DC}$ and *CBM* directly modify the probabilities without introducing additional parameters, while *CC* and *Penalty* use specific calibration parameters. In zero-shot calibration, these parameters are set to prior probabilities. We will now investigate whether the few-shot training of the calibration parameters further improves the performance.

We observe that training the calibration parameters on just a few samples further enhances the performance of the calibrated systems. Algorithm 1 presents the process of few-shot training of penalty calibration used in our few-shot investigation.

Algorithm 1 Few-Shot Training of Penalty Calibration

```
Input: Few-shot training samples $\boldsymbol{D} = \{(x, y)\}$, initial calibration parameters $\boldsymbol{p}$, number of
    epochs $E$, Learning rate $\eta$
Output: Trained parameters $\boldsymbol{p}$
for t $= 1$ to $E$ do
    for each $(x, y)$ in $\boldsymbol{D}$ do
        $\boldsymbol{l} = get\_probs(x) - \boldsymbol{p}$
        $\hat{y} \leftarrow \arg\max(\boldsymbol{l})$
        if $y \neq \hat{y}$ then
            $\boldsymbol{p}[\hat{y}] \leftarrow \boldsymbol{p}[\hat{y}] + \eta$
            $\boldsymbol{p}[y] \leftarrow \boldsymbol{p}[\hat{y}] - \eta$
        end if
    end for
end for
```

Figure 3.2 shows the results for the RoBERTa model on the AG News and Amazon Polarity tasks. We also compare calibration methods in few-shot scenarios with the NLI-based zero-shot classification baseline proposed by Yin et al. (2019). Table 3.5 shows the complete results of few-shot calibration.

BERT$_{\text{Base}}$

		AG News		Amazon-P		Pawsx		XNLI		Avg	
nli-based ZR		54.9		**82.3**		48.2		34.8		55.1	
calibration		Penalty	CC	Penalty	CC	Penalty	CC	Penalty	CC	Penalty	CC
zero-shot	0	67.9	74.6	61.7	61.7	45.4	45.4	42.6	41.4	54.4	55.8
few-shot	1	$65.6_{3.8}$	$75.7_{1.0}$	$67.8_{7.6}$	$71.0_{5.6}$	$51.1_{0.9}$	$\underline{51.4_{0.9}}$	$42.0_{1.8}$	$41.2_{1.9}$	$56.6_{3.5}$	$59.8_{2.4}$
	2	$67.2_{3.1}$	$75.9_{1.6}$	$71.9_{4.4}$	$\underline{72.2_{3.2}}$	$51.0_{1.1}$	$50.7_{1.0}$	$42.7_{0.6}$	$42.5_{0.9}$	$58.2_{2.3}$	$\underline{\mathbf{60.3_{1.7}}}$
	4	$67.9_{3.9}$	$76.6_{0.7}$	$73.4_{3.8}$	$70.3_{2.9}$	$\underline{\mathbf{51.6_{1.3}}}$	$50.9_{1.3}$	$42.8_{0.6}$	$\underline{42.8_{0.3}}$	$58.9_{2.4}$	$60.2_{1.3}$
	8	$69.1_{1.5}$	$\underline{76.9_{0.1}}$	$75.2_{2.3}$	$71.8_{1.2}$	$\underline{\mathbf{51.6_{1.1}}}$	$49.9_{0.6}$	$\underline{\mathbf{42.9_{0.2}}}$	$42.7_{0.2}$	$59.7_{1.3}$	$\underline{\mathbf{60.3_{0.5}}}$
	16	$\underline{69.6_{1.7}}$	$\underline{\mathbf{76.9_{0.1}}}$	$\underline{76.0_{1.0}}$	$71.4_{1.2}$	$51.4_{1.1}$	$49.7_{1.0}$	$42.8_{0.3}$	$42.6_{0.2}$	$\underline{60.0_{1.0}}$	$60.2_{0.6}$

RoBERTa$_{\text{Base}}$

		AG News		Amazon-P		Pawsx		XNLI		Avg	
nli-based ZR		67.9		84.8		45.3		34.3		58.1	
calibration		Penalty	CC	Penalty	CC	Penalty	CC	Penalty	CC	Penalty	CC
zero-shot	0	75.6	74.1	79.5	79.5	45.4	45.4	41.4	39.8	60.5	59.7
few-shot	1	$75.6_{2.6}$	$77.2_{1.5}$	$77.4_{8.0}$	$\underline{81.3_{4.9}}$	$48.4_{1.8}$	$48.4_{1.4}$	$45.9_{0.9}$	$44.8_{1.5}$	$61.8_{3.3}$	$62.9_{2.3}$
	2	$73.9_{2.8}$	$77.3_{1.2}$	$81.6_{4.3}$	$80.8_{2.4}$	$49.0_{1.6}$	$48.3_{0.9}$	$46.3_{0.7}$	$45.8_{0.7}$	$62.7_{2.4}$	$\underline{63.1_{1.3}}$
	4	$74.5_{1.9}$	$77.6_{1.0}$	$82.2_{4.4}$	$79.6_{1.6}$	$49.3_{0.6}$	$\underline{48.5_{0.9}}$	$\underline{\mathbf{47.2}_{0.2}}$	$\underline{46.0_{0.3}}$	$63.3_{1.8}$	$62.9_{1.0}$
	8	$76.6_{1.1}$	$78.1_{0.5}$	$\underline{\mathbf{85.2_{1.0}}}$	$79.7_{1.5}$	$\underline{\mathbf{49.6_{0.4}}}$	$48.1_{0.7}$	$47.1_{0.3}$	$\underline{46.0_{1.0}}$	$64.6_{0.7}$	$63.0_{0.9}$
	16	$\underline{78.3_{0.5}}$	$\underline{\mathbf{78.4_{0.3}}}$	$85.1_{1.0}$	$79.7_{1.6}$	$49.4_{0.6}$	$48.1_{0.4}$	$47.0_{0.2}$	$\underline{46.0_{0.9}}$	$\underline{\mathbf{65.0_{0.6}}}$	$\underline{63.1_{0.8}}$

Table 3.5: Results of few-shot calibration. *nli-based ZR* refers to the NLI-based zero-shot classification baseline (Yin et al., 2019).

Prior research (Zhao and Schütze, 2021) showed that few-shot learning can be unstable due to the randomness. However, our experimental results indicate that the variation in performance diminishes obviously only after the number of shots reaches four.

3.1.4.2 Results on Multilingual Encoders

Our experimental results on multilingual datasets demonstrate that calibration methods are also effective for multilingual encoders (Table 3.6).

Our experiments cover a large range of languages, considering both language availability, i.e., if or how much language data exists in the pretraining corpus, and language diversity, i.e., to which language family a language belongs. Specifically, for Amazon-S, XNLI, and PAWS-X, we use the original test sets, mainly containing the high-resource languages. In the multilingual AG News task, we include many low-resource and unseen languages by generating parallel multilingual test sets using machine translation techniques. Recent research by Hu et al. (2020b) and Liu et al. (2022d) shows that automatically translated test sets are useful for measuring cross-lingual performance. Hence, we adopt their methodology and expand the language coverage of the AG News dataset to 25. Table 3.7 provides an overview of languages covered by the multilingual AG News dataset.

The results on multilingual BERT and XLM-R show that all four calibration methods improve the multilingual performance averaged across all tasks. For both models, *CBM* always emerges as the top-performing approach. Different from other approaches that predict the label with one input by another, *CBM* considers all other input samples in the test set when making each individual prediction. This could account for the substantial advantage of *CBM* over the others in terms of performance.

	AG News	Amazon-S	XNLI	PAWS-X	Avg.
mBERT$_{\text{Base}}$					
+ *no calib.*	32.8	20.5	33.6	33.9	30.2
+ *PMI*$_{DC}$	48.8	22.5	33.6	44.4	37.3
+ *CBM*	53.8	**25.1**	34.9	**49.2**	**40.8**
+ *CC (max)*	53.9	23.9	35.1	44.8	39.4
+ *Penalty (max)*	**54.6**	23.8	**35.3**	47.1	40.2
XLM-R$_{\text{Base}}$					
+ *no calib.*	45.4	21.9	35.0	31.7	33.5
+ *PMI*$_{DC}$	59.8	23.0	33.6	37.8	38.6
+ *CBM*	**63.3**	**28.9**	**37.8**	**46.3**	**44.1**
+ *CC (max)*	59.6	23.7	35.3	43.7	40.6
+ *Penalty (max)*	57.5	23.6	35.8	43.4	40.1

Table 3.6: Results of calibration methods on multilingual datasets. We report the best results for *CC* and *Penalty* in different few-shot settings.

Code	Languages	Language Accessibility	Language Family
af	Afrikaans	Low-resource	Indo-European
co	Corsican	Unseen languages	Indo-European
eo	Esperanto	Unseen languages	Artificial
haw	Hawaiian	Unseen languages	Austronesian
hmn	Hmong	Unseen languages	Sino-Tibetan
ht	Haitian Creole	Low-resource	Indo-European
ig	Igbo	Unseen languages	Niger-Congo
jw	Javanese	Low-resource	Austronesian
km	Khmer	Unseen script	Austronesian
mi	Maori	Low-resource	Austronesian
mn	Mongolian	Low-resource	mongolian
mt	Maltese	Unseen languages	Afro-Asiatic
my	Burmese	Low-resource	Sino-Tibetan
ny	Chichewa	Unseen languages	Niger-Congo
or	Odia	Unseen script	Indo-European
sm	Samoan	Unseen languages	Austronesian
sn	Shona	Unseen languages	Dravadian
st	Sesotho	Unseen languages	Dravadian
sw	Swahili	Low-resource	Dravadian
ta	Tagalog	Low-resource	Austronesian
te	Telugu	Low-resource	Dravadian
tl	Tamil	Low-resource	Dravadian
ug	Uighur	Unseen languages	Turkic
ur	Urdu	Low-resource	Indo-European
uz	Uzbek	Low-resource	Turkic
zu	Zulu	Unseen languages	Niger-Congo

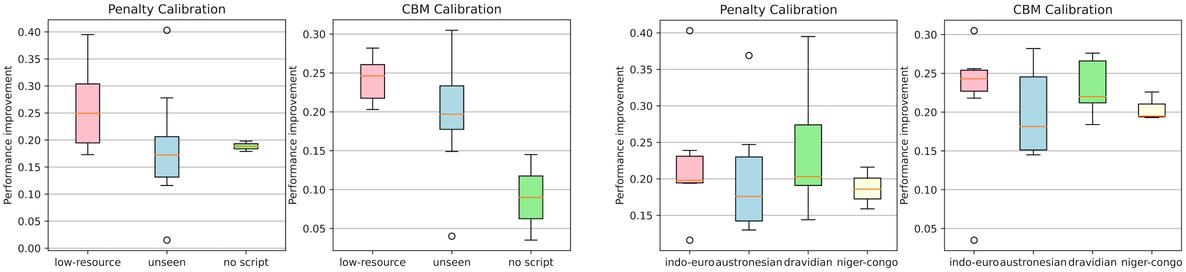
Table 3.7: Overview of languages covered by the multilingual AG News dataset.

3.1.4.3 Multilingual Analysis

Now we analyze how different language properties correlate with the performance of multilingual BERT on the AG News task. Table 3.8 shows the complete results of mBERT on the multilingual AG News dataset across all 25 languages.

	af	co	en	eo	haw	hmn	ht	ig	jw	km	mi	mn	mt	my
No calib.	40.4	32.6	47.3	31.9	27.1	30.9	35.7	30.2	38.0	33.3	29.0	32.0	29.9	33.8
Penalty	64.3	44.2	69.6	72.3	40.1	49.6	55.2	48.8	62.6	51.2	46.3	62.2	57.6	64.7
CBM	64.7	58.3	69.1	62.4	42.0	50.8	60.9	49.6	63.9	47.8	49.5	53.0	57.2	54.1
CC	65.6	59.7	67.8	68.0	43.4	49.7	65.2	52.4	66.4	41.4	51.2	55.4	57.4	51.7
PMI_{DC}	60.2	35.3	60.0	61.7	35.9	33.5	33.5	49.2	61.5	42.2	49.6	54.7	61.1	47.6
	ny	**or**	**sm**	**sn**	**st**	**sw**	**ta**	**te**	**tl**	**ug**	**ur**	**uz**	**zu**	**avg.**
No calib.	29.8	25.4	30.3	32.2	30.4	33.4	28.8	32.5	42.6	25.5	33.2	33.9	34.5	32.8
Penalty	51.4	45.2	43.5	52.4	44.8	72.9	65.6	59.9	61.7	27.0	52.6	59.1	50.3	54.6
CBM	52.4	28.9	46.1	53.4	48.8	59.9	57.0	60.0	64.6	29.5	56.8	58.9	53.7	53.8
CC	51.2	28.7	47.5	52.5	49.1	64.1	56.5	52.4	62.6	27.9	53.1	60.3	49.6	53.7
PMI_{DC}	50.2	28.6	43.9	50.9	44.6	61.6	50.1	43.6	66.1	29.3	55.0	56.4	51.3	48.8

Table 3.8: Results of mBERT on the multilingual AG News dataset across all languages.



(a) Language Accessibility

(b) Language Diversity

Figure 3.3: Performance improvement of multilingual BERT with two calibration methods.

Language Accessibility. We first group the evaluation languages into low-resource languages, unseen languages, and languages with unseen scripts to determine the influence of language accessibility. Low-resource languages are languages that are contained in the pretraining corpus, but only account for a small amount of it. Unseen languages do not occur in the pretraining. Thus, the multilingual encoder has never seen them. The hardest cases are languages with unseen scripts, where the model has not even encountered the characters of the language. However, in our test set, they are not strictly languages with unseen scripts because of the frequently occurring code-switching led by machine translation. Figure 3.3 (a) shows that low-resource languages perform generally better than the other two types of unseen languages, indicating that the multilingual encoder's access to languages in the pretraining is crucial for the performance enhancement via calibration.

Language Diversity. We further group the languages according to their phylogenetic relationships, i.e., from which language family they are. We analyze the language families containing at least 3 languages. The box plots in Figure 3.3 (b) reveal that the impact of calibrating multilingual encoders varies across different language groups. Specifically, we observe that Indo-European and Dravidian languages tend to benefit more from calibration than Austronesian and Niger-Congo languages.

This discrepancy suggests that the effectiveness of calibration techniques can be influenced by the language accessibility of multilingual encoders and the linguistic characteristics of language families.

3.1.5 Sum-Up

In conclusion, in this subsection, we focus on boosting the zero-shot potential of multilingual encoders through probability calibration. We address the bias issue in the mask token prediction of label words by introducing various calibration techniques that modify the probabilities of these words. By applying these methods, we achieve substantial performance gains across a diverse range of tasks on both monolingual and multilingual encoders. Notably, with a minimal number of training examples, the calibrated probabilities yield significant enhancements.

We propose a simple yet effective calibration method to enhance the zero-shot performance for monolingual and multilingual encoders. While our work shows the effectiveness of calibration for enhancing the prediction with multilingual tasks, it is important to note that our research is primarily focused on classification tasks with multilingual encoders. As a result, our findings and proposed methods may not directly translate to generation tasks, such as question answering (QA), which involve the use of generative multilingual models. Future investigations should explore the application of our calibration methods on generation tasks and evaluate their effectiveness in enhancing the performance of generative multilingual models. This extension could provide valuable insights into the potential benefits and limitations of our approaches across a broader range of NLP tasks.

3.2 Prompt Augmented by Cross-Lingual Retrieval for Low-Resource Languages

This section corresponds to the following work:

> **Ercong Nie***, Sheng Liang*, Helmut Schmid, and Hinrich Schütze. 2023. Cross-Lingual Retrieval Augmented Prompt for Low-Resource Languages. In Findings of the Association for Computational Linguistics: ACL 2023, pages 8320–8340, Toronto, Canada. Association for Computational Linguistics.
> * equal contributions.

Declaration of Co-Authorship. Sheng Liang and I conceived the idea of using cross-lingual retrieval samples to augment the zero-shot cross-lingual understanding of Multilingual Pretrained Language Models (MPLMs) through prompt-based learning. I implemented the idea and conducted all the experiments. Sheng Liang and I completed the manuscript together. Helmut Schmid and Hinrich Schütze supervised this project by providing feedback and revising the manuscript.

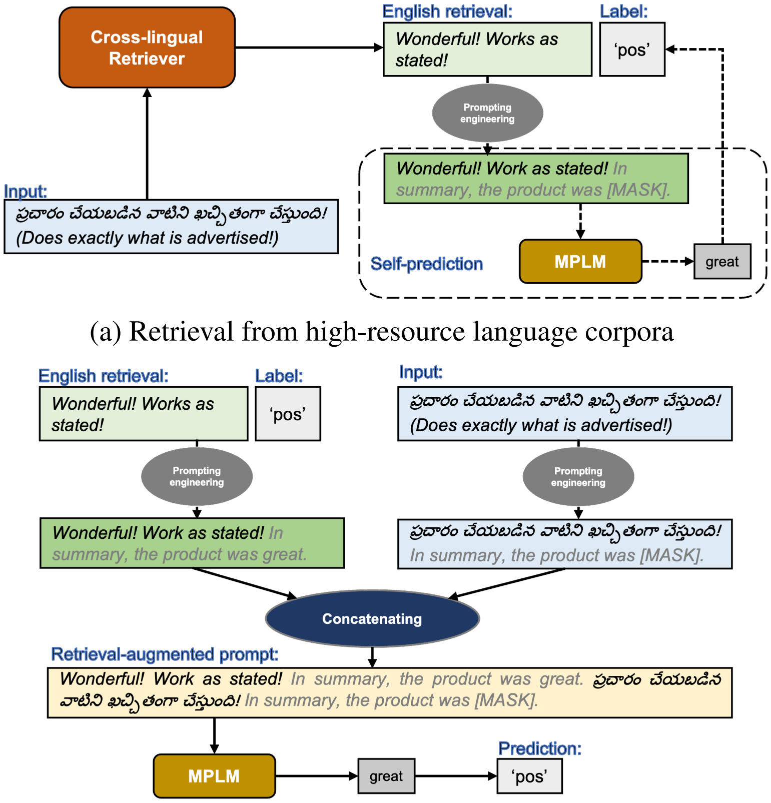
Summary of This Section

Multilingual Pretrained Language Models (MPLMs) perform strongly in cross-lingual transfer with zero-shot prompt-based learning. In this section, we propose **P**rompts **A**ugmented by **R**etrieval **C**ross-lingually (**PARC**) to improve zero-shot performance on low-resource languages (LRLs) by augmenting the context with prompts consisting of semantically similar sentences retrieved from a high-resource language (HRL). PARC improves zero-shot performance on three downstream tasks (sentiment classification, topic categorization, natural language inference) with multilingual parallel test sets across 10 LRLs covering 6 language families in unlabeled (+5.1%) and labeled settings (+16.3%). PARC also outperforms fine-tuning by 3.7%. We find a significant positive correlation between cross-lingual transfer performance on one side, and the similarity between high- and low-resource languages, as well as the amount of low-resource pretraining data on the other side. A robustness analysis suggests that PARC has the potential to achieve even stronger performance with more powerful MPLMs.

3.2.1 Background and Overview of PARC

Multilingual pretrained language models (MPLMs) (Devlin et al., 2019; Conneau et al., 2020; Liu et al., 2020; Xue et al., 2021; Shliazhko et al., 2022), pretrained on multilingual corpora with $>$100 languages, exhibit strong multilinguality on downstream tasks (Hu et al., 2020b). Low-resource languages (LRLs), for which little text data is available for pretraining monolingual pretrained language models (PLMs), benefit from MPLMs. However, the lack of LRL data leads to an imbalanced language distribution in the pretraining corpora of MPLMs (Wu and Dredze, 2020). LRLs are therefore underrepresented in pretraining, resulting in bad performance. Furthermore, the scarcity of domain- or task-specific annotated data of LRLs makes it difficult to apply the pretraining-fine-tuning paradigm to LRLs (Lauscher et al., 2020). Given that the pretraining-fine-tuning paradigm always has a high demand for domain-specific labeled data, another line of research—prompt-based learning—emerges, focusing on exploiting large pretrained language models by reformulating the input. The prompt is designed to help PLMs “understand” the task better and “recall” what has been learned during the pretraining. In particular, Brown et al. (2020) propose a simple in-context learning approach without any fine-tuning, which adds training examples as additional context to test examples. Instead of using random examples as context, KATE (Liu et al., 2022a) and SOUP (Liu et al., 2022e) retrieve semantically similar examples as prompt for monolingual in-context learning. The above-mentioned prompt-based learning techniques require no parameter updating, while there is also work employing sampled similar examples for prompt-based fine-tuning (Gao et al., 2021). Unlike Brown et al. (2020), who created prompts with manually selected examples, these approaches augment the context by retrieving related information from external corpora, allowing the PLMs to capture more domain- or task-specific knowledge. The prompt-based method offers a new form of zero-shot or few-shot learning in multilingual NLP studies. It involves performing a specific task using prompts, without labeled data in the target language, and has the potential of being an effective method for LRLs lacking annotated data.

Our work improves the zero-shot transfer learning performance of LRLs on three differ-


Cross-lingual Retriever
English retrieval:
Wonderful! Works as stated!
Label:
'pos'
Prompting engineering
Wonderful! Work as stated! In summary, the product was [MASK].
Input:
ప్రచారం చేయబడిన వాటిని ఖచ్చితంగా చేస్తుంది!
(Does exactly what is advertised!)
Self-prediction
MPLM
great
English retrieval:
Wonderful! Works as stated!
Label:
'pos'
Input:
ప్రచారం చేయబడిన వాటిని ఖచ్చితంగా చేస్తుంది!
(Does exactly what is advertised!)
Prompting engineering
Prompting engineering
Wonderful! Work as stated! In summary, the product was great.
ప్రచారం చేయబడిన వాటిని ఖచ్చితంగా చేస్తుంది! In summary, the product was [MASK].
Concatenating
Retrieval-augmented prompt:
Wonderful! Work as stated! In summary, the product was great. ప్రచారం చేయబడిన వాటిని ఖచ్చితంగా చేస్తుంది! In summary, the product was [MASK].
MPLM
great
Prediction:
'pos'

(a) Retrieval from high-resource language corpora

(b) Prediction with a retrieval-augmented prompt

Figure 3.4: Main idea of PARC: we enhance zero-shot learning for low-resource languages (LRLs) by cross-lingual retrieval from **labeled/unlabeled** high-resource languages (HRLs). (a) An LRL input sample is taken as a query by the cross-lingual retriever to retrieve the semantically most similar HRL sample from the HRL corpus. The label of the retrieved HRL sample is obtained either from the corpus (**labeled** setting) or by self-prediction (**unlabeled** setting). (b) The retrieved HRL sample, together with its label and the input sample, is reformulated as prompts. The cross-lingual retrieval-augmented prompt is created by concatenation and taken by the MPLM for prediction. Our experiments show that PARC outperforms other zero-shot methods and even fine-tuning.

ent classification tasks by taking advantage of cross-lingual information retrieval and the multilinguality of MPLMs. Specifically, we retrieve semantically similar cross-lingual sentences as prompts and use the cross-lingual retrieval information to benefit the LRLs from the multilinguality of MPLMs and achieve better performance in the zero-shot setting[1].

Our main contributions are: (1) We propose **P**rompts **A**ugmented by **R**etrieval **C**rosslingually (**PARC**), a pipeline for integrating retrieved cross-lingual information into prompt engineering for zero-shot learning (Figure 3.4). (2) We conduct experiments on several multilingual tasks,

[1] Different from the zero-shot cross-lingual transfer learning where MPLMs are finetuned on HRLs (Hu et al., 2020b), our zero-shot setting does not involve fine-tuning. Details in §3.2.5.4

showing that PARC improves the zero-shot performance on LRLs by retrieving examples from both labeled and unlabeled HRL corpora. (3) To find an optimal configuration of our PARC pipeline, we conduct a comprehensive study on the variables that affect the zero-shot performance: the number of prompts, the choice of HRL, and the robustness w.r.t. other retrieval methods and MPLMs.

3.2.2 Methodology

This work aims to improve the performance of MPLMs on LRLs in the zero-shot setting by leveraging retrieved cross-lingual contents from HRLs. For that, we design the PARC pipeline that can be applied to labeled and unlabeled scenarios, i.e., the HRL information can be retrieved from either labeled or unlabeled corpora.

As Figure 3.4 shows, the PARC pipeline consists of two steps: (a) Cross-lingual retrieval from high-resource language corpora, and (b) prediction with a retrieval-augmented prompt. Figure 3.4 shows an example: A Telugu input sentence from a sentiment classification task is first fed into the cross-lingual retriever to fetch the semantically closest sample from the HRL corpus, i.e., English in this case. In the second step, the retrieved HRL sample, together with its label and the LRL input sentence, is transformed into a prompt. For prompt-based classification, we need (i) a *pattern* which converts the input sentence into a cloze-style question with a mask token, and (ii) a representative word (called *verbalizer*) for each possible class. Converting the classification task into a cloze-style question aligns seamlessly with the framework of our proposed PARC method, because it not only performs zero-shot learning well but, more significantly, facilitates better integration of the retrieved cross-lingual contexts.

In our example, we use the pattern $P(X) = X \circ$ “`In summary, the product was [MASK].`” to convert the retrieved English sentence into “`Wonderful! Works as stated! In summary, the product was [MASK].`”, where $\circ$ is the string concatenation operator. A verbalizer such as {`pos` → “great”, `neg` → “terrible”}, which maps the original labels {`pos`, `neg`} onto words in the vocabulary, is then used to replace the `[MASK]` token with the verbalized label word “great”, standing for the correct label `pos` of this sentence. We call the resulting English sentence (in our example: “`Wonderful! Works as stated! In summary, the product was great.`”) the “cross-lingual context”. At last, we fill in the same pattern with the input Telugu sentence and append it to the cross-lingual context. We feed this cross-lingual retrieval augmented input to the MPLM. The MPLM returns for each of the verbalizers its probability of being the masked token.

More formally, let $X_i^L \in D^L$ be an input sample from the LRL test set, $(X_j^H, y_j) \in D_{lb}^H$ and $X_j^H \in D_{un}^H$ denote the HRL data from the *labeled* and *unlabeled* corpora, respectively, where X_j is the text sample and y_j its class label from a label set Y. As Eq. (3.1) shows, the cross-lingual retriever CLR takes the HRL corpora D^H and a given LRL input sentence X_i^L. It returns an ordered list of HRL sentences D^{R_i} according to the semantic similarity. We then have $(X_k^{R_i}, y_k^{R_i}) \in D_{lb}^{R_i}$ and $X_k^{R_i} \in D_{un}^{R_i}$ for labeled and unlabeled scenarios, respectively, where $X_k^{R_i}$ is the k-th most similar HRL sentence to the LRL input X_i^L.

$$D^{R_i} = CLR(X_i^L, D^H) \tag{3.1}$$

			En	Af	Jv	Mn	My	Sw	Ta	Te	Tl	Ur	Uz	Avg
MAJ			25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
Direct			52.5	41.8	27.4	42.5	32.2	31.3	31.5	33.0	31.6	46.9	44.8	36.3
UN	k=1		53.7	52.8	46.2	46.5	46.1	42.8	43.3	44.3	45.0	51.0	49.7	46.7
	k=3	BoR	55.8	53.6	46.2	47.1	48.2	44.9	44.5	46.3	47.1	52.6	51.0	48.1
		CONC	53.5	52.4	45.9	44.9	44.8	42.9	41.7	46.6	46.0	52.0	51.6	46.9
	k=5	BoR	57.1	54.4	47.0	47.0	48.0	46.6	44.8	45.8	48.5	53.1	52.3	48.7
		CONC	53.5	48.0	38.2	41.3	36.3	36.9	39.5	41.4	42.9	50.5	49.6	42.4
	k=10	BoR	57.5	55.3	46.3	46.4	47.6	45.6	44.1	46.7	47.7	53.0	51.4	48.4
		CONC	46.4	41.1	36.2	38.3	36.6	34.9	34.6	35.8	40.7	46.3	45.0	38.9
	k=20	BoR	59.7	57.2	48.1	46.7	50.0	47.9	46.0	48.9	49.6	55.4	53.2	50.3
		CONC	50.0	48.4	42.3	41.4	43.3	43.1	39.3	44.3	48.1	47.9	48.4	44.6
	k=30	BoR	60.1	57.4	49.0	47.4	51.1	49.2	47.1	48.7	50.1	56.5	54.4	51.1
		CONC	50.7	47.6	43.9	38.2	42.9	42.5	41.8	44.5	47.7	47.1	47.3	44.3
LB	k=1		74.9	75.4	68.1	63.5	68.2	64.0	62.8	65.6	64.8	72.5	71.4	67.6
	k=3	BoR	77.1	77.1	69.6	65.6	71.1	67.6	65.6	68.4	65.9	74.6	74.4	70.0
		CONC	75.6	74.8	67.3	63.1	60.3	59.0	60.5	67.1	65.9	73.3	72.4	66.4
	k=5	BoR	78.1	78.6	69.0	64.4	72.9	68.8	65.9	69.3	66.4	75.8	75.4	70.6
		CONC	74.6	66.5	48.2	53.9	44.9	45.4	52.1	59.5	56.0	70.9	63.6	56.1
	k=10	BoR	78.7	79.4	70.5	67.0	72.9	68.3	66.6	70.7	67.2	76.6	75.9	71.5
		CONC	61.2	52.7	43.2	48.0	44.5	42.5	41.3	45.0	50.1	62.3	56.7	48.6
	k=20	BoR	79.0	79.7	70.7	67.5	72.5	70.0	67.5	70.7	68.1	77.4	76.3	72.0
		CONC	67.4	65.1	55.8	55.6	57.6	58.3	51.2	61.0	62.8	66.4	66.0	60.0
	k=30	BoR	79.0	79.7	71.3	67.6	72.8	69.9	68.1	71.1	69.4	77.2	76.7	72.4
		CONC	72.8	71.1	62.1	57.0	61.6	60.4	57.9	67.9	64.6	71.6	69.3	64.3

Table 3.9: Results of topic categorization task on AG News Dataset. k is the number of retrieved cross-lingual samples. MAJ is the majority baseline. Avg is the average accuracy across 10 LRLs. En is the HRL for retrieval. BoR refers to the *Bag of Retrieval* strategy, CONC refers to the *Concatenation* strategy.

The prompt pattern $P(.)$ converts an HRL input sentence $X_k^{R_i}$ into a cloze-style form with a mask token. The verbalizer $v(.)$ is a bijective mapping from the set of class labels Y to a set of verbalized words V from the HRL vocabulary. We use the verbalized label word to fill in the mask token in the prompt pattern, and construct the cross-lingual context C_k^i for the input X_i^L with the k-th most similar HRL sample $X_k^{R_i}$:

$$C_k^i = P(X_k^{R_i}, v(y_k^{R_i})) \tag{3.2}$$

The cross-lingual context C_k^i is then concatenated with the prompted LRL input as the input I to the MPLM:

$$I_i = C_k^i \circ P(X_i^L) \tag{3.3}$$

The MPLM M performs masked token prediction and returns the probabilities $p = M(I_i)$ of all candidate words for the masked token in I_i. We predict the class $\hat{y}$ whose verbalizer $v(\hat{y})$ received the highest probability from model M:

$$\hat{y} = \arg\max_{y \in Y} p(v(y)) \tag{3.4}$$

In the labeled scenario, we obtain the correct label $y_k^{R_i}$ of the HRL sentence from $D_{lb}^{R_i}$. In the unlabeled scenario, we predict the label using the same prompt-based classification method without cross-lingual context, similar to Eq. (3.4). We call this the *self-prediction* method:

$$\hat{y}_k^{R_i} = \arg\max_{y \in Y} M(P(X_k^{R_i}), v(y)) \tag{3.5}$$

In order to use more cross-lingual information, we retrieve the K most similar HRL samples. With each sample, we obtain verbalizer probabilities as described above and retrieve the class whose verbalizer has the largest sum of probabilities. We call this method the Bag-of-Retrieval (BoR) strategy. We also tried concatenating the different cross-lingual contexts (CONC method), but the resulting performance has been worse (see Table 3.9).

3.2.3 Experimental Setup

3.2.3.1 Datasets

Base Datasets Three representative classification tasks are selected for evaluation in this work: binary sentiment analysis on Amazon product reviews (Keung et al., 2020), topic classification on AG News texts (Zhang et al., 2015), and natural language inference on XNLI (Conneau et al., 2018).

Amazon Reviews dataset categorizes the shopping reviews into 5-star ratings from 1 to 5. In order to satisfy a binary classification setting, we select the reviews with rating 1 as `negative` (0) and 5 as `positive` (1) for our experiments. The following pattern $P(X)$ and verbalizer v are defined for an input review text X:

- $P(X) = X \circ$ "All in all, it was [MASK]."
- $v(0) =$ "terrible", $v(1) =$ "great"

AG News is a collection of more than 1 million news articles for topic classification. The news topic categories contained in the dataset are `World` (0), `Sports` (1), `Business` (2), and `Tech` (3). The pattern and verbalizers are as follows:

- $P(X) =$ "[MASK] News: " $\circ X$
- $v(0) =$ "World", $v(1) =$ "Sports",
  $v(2) =$ "Business", $v(3) =$ "Tech"

XNLI is a multilingual version of the MultiNLI dataset (Williams et al., 2018). We use a subset of the original XNLI dataset in our experiment. The text in each data item consists of two parts. Sentence A is the premise, and sentence B is the hypothesis. The NLI task is to predict the type of inference between the given premise and hypothesis among the three types: `entailment` (0), `neutral` (1), and `contradiction` (2). For a given sentence pair X_1 and X_2, we design the pattern and verbalizer as:

- $P(X_1, X_2) = X_1 \circ$ "? [MASK]," $\circ X_2$
- $v(0) =$ "Yes" , $v(1) =$ "Maybe" , $v(2) =$ "No"

Construction of Multilingual Parallel Test Sets Parallel test datasets for evaluating cross-lingual transfer performance on LRLs are rare. However, recent research conducted by Hu et al. (2020b); Liu et al. (2022d) shows that automatically translated test sets are useful for measuring cross-lingual performance. Hence, we adopt their methodology and construct datasets for different tasks by automatically translating English test sets to targeted LRLs. We use the Python API of the Google Translate System to implement the construction of multilingual parallel test sets in our experiment. We also validate the translation effectiveness and quality. The original XNLI datasets include two low-resource languages that are used in our experiments (Swahili and Urdu), so we use them as the "gold" standard for our translation validation.

We compare the cross-lingual transfer performance on translation test sets and original test sets of XNLI. We also measure the translation quality by using the original sets as the gold standard. Through the validation conducted on these two languages within the XNLI task, we infer that the translation method is effective and could be generalized to other languages and tasks.

In our experiment, we use multilingual parallel test sets created by machine translation from English to target low-resource languages. To explore the effect of machine translation-created test sets, we compare the cross-lingual transfer performance on translation test sets and original test sets of XNLI. The original XNLI datasets include two low-resource languages that we used in our experiments, i.e., Swahili (sw) and Urdu (ur). We also measure the translation quality by using the original sets as the gold standard. The analysis results (Table 3.10) suggest that machine-translated test sets are useful as a proxy for evaluating cross-lingual performance on LRLs.

Languages		sw	ur
Performance	MT Acc.	34.00	33.92
	OV Acc.	34.07	33.87
	Diff	0.07	-0.05
	P-Value	0.85	0.92
Translation Quality	BLEU	56.39	64.96
	chrF	49.58	59.89
	Sim.	81.82	81.19

Table 3.10: Comparison of performance on machine translation-created XNLI test sets (MT) and the original version of XNLI test sets (OV) in sw and ur languages. BLEU & chrF scores and semantic similarities (Sim.) are computed to measure the translation quality of machine translation-created test sets.

Following Wu and Dredze (2020), we regard languages with a WikiSize[2] of less than 7 as LRLs. We construct a test set consisting of 10 LRLs in 6 language families: Indo-European (Afrikaans - af, Urdu - ur), Austronesian (Javanese - jv, Tagalog - ta), Altaic (Mongolian - mn, Uzbek - uz), Dravidian (Tamil - tl and Telugu - te), Sino-Tibetan (Burmese - my), and Niger-Congo (Swahili - sw). Table 3.11 shows more information of the test sets.

[2] WikiSize less than 7 means that the Wikipedia corpus of the language is smaller than 0.177 GB.

Task	Dataset	Size	#Label	Languages
Sentiment Analysis	Amazon Reviews	1000	2	af, ur, jv, ta, mn, uz, tl, te, mn, sw
Topic Categorization	AG News	2000	4	
Sentence Pair Classification	XNLI	1500	3	

Table 3.11: Overview of the test sets for the three tasks. Size refers to the number of samples for each LRL.

HRL Corpora To retrieve rich and diverse information, a large-scale general corpus or knowledge base in the different HRLs would be the ideal sentence retrieval pool. In practice, however, a trade-off is necessary in order to save computational resources. Following Wang et al. (2022a), we therefore use the task-specific labeled training set of English as the sentence pool in our experiments. The selection of the HRL will be discussed in §3.2.5.2.

3.2.3.2 Baseline

We compare PARC with the following baselines in both labeled and unlabeled settings:

MAJ: The majority baseline. Since we construct the test sets to be balanced, MAJ is equivalent to a random guess.

Random: We randomly retrieve a cross-lingual sentence as prompt, similar to the simple in-context learning using examples without semantic similarity to the input (Brown et al., 2020).

Direct: The pattern filled with the input sample is directly fed to the MPLM for prediction, without adding cross-lingual context to the prompts.

Finetune: The MPLM is first finetuned with the retrieved high-resource sentences. Then the low-resource test input is predicted by the finetuned MPLM. We use the Cross Entropy Loss as the objective function for fine-tuning and AdamW for optimization with a learning rate of 1e-5. Since the fine-tuning data is very limited, we only train for a single epoch to avoid overfitting.

Our test sets are constructed by machine translation. Therefore, we cannot apply a translation baseline, where we translate the input sample into the high-resource language before feeding it to the MPLM. The Appendix presents a different experiment where we compare with a translation baseline.

3.2.3.3 Models

Cross-Lingual Retriever The retrieval methods used in monolingual NLP are either based on sparse or dense representations. Sparse representations such as BM25 (Manning et al., 2008), which is based on term frequency, cannot be used for cross-lingual retrieval as the shared words across different languages are normally scarce. Therefore dense representations from deep learning methods such as LASER (Artetxe and Schwenk, 2019) and sentence-BERT (Reimers and Gurevych, 2019) are more suitable for our pipeline.

We choose the multilingual sentence transformer (Reimers and Gurevych, 2020) version "*paraphrase-multilingual-mpnet-base-v2*" as the retriever in our experiments. This multilingual

	Amazon	**AGNews**	**XNLI**	**Avg.**
MAJ	50.0	25.0	33.3	36.1
Random	48.2	25.6	32.4	35.4
Direct	53.8	36.3	33.1	41.1
Finetune	68.6	57.9	34.5	53.7
PARC-unlabeled	58.4	46.7	33.5	46.2
PARC-labeled	**68.9**	**67.6**	**35.8**	**57.4**

Table 3.12: Overview of results on three classification tasks. The reported numbers are averaged across 10 evaluation LRLs. The number of prompts k is 1 in the relevant baselines and our methods for all three tasks.

retriever is based on XLM-R (Conneau et al., 2020) and trained on parallel data from 50+ languages by employing knowledge distillation. Through the multilingual sentence transformer, sentences are represented by embeddings. We use the sentence embeddings to calculate the cosine similarity between the LRL inputs and HRL sentences and rank the most similar ones for retrieval. Robustness with respect to other cross-lingual retrievers will be discussed in §3.2.5.3.

Multilingual Pretrained Language Model In order to solve cloze-style classification tasks, we use the pretrained multilingual BERT model "*bert-base-multilingual-cased*" (Devlin et al., 2019). It contains 178M parameters and was trained on Wikipedia corpora in 104 languages. In §3.2.5.3, we will also explore XLM-R (Conneau et al., 2020), another multilingual pretrained language model.

All the models mentioned above were implemented using the Huggingface Transformers library (Wolf et al., 2020).

3.2.4 Results

Table 3.12 presents an overview of the results on the three tasks[3]. PARC outperforms the *MAJ*, *Direct*, and *Random* baseline on all three tasks, in both labeled and unlabeled settings: When retrieving from unlabeled high-resource language corpora, the performance is improved by **4.6%**, **10.4%** and **0.4%** compared to *Direct* on Amazon Review, AG News, and XNLI, respectively. When retrieving from labeled HRL corpora, the performance is improved by **15.1%**, **31.3%**, and **2.7%**. The *Finetune* baseline uses the label of retrieved examples for prompt-based fine-tuning. Hence, it is fair to compare it with *PARC* in the labeled setup rather than the unlabeled one. *PARC-labeled* outperforms *Finetune* by **0.3%**, **9.7%** and **1.3%** on the three tasks respectively.

Although our proposed methods perform better than the baselines on all three tasks, the degree of improvement differs. A large improvement is found on AG News, the topic categorization task, while XNLI only shows a slight improvement. An explanation for this could be that the natural language inference task is more difficult than topic categorization, especially in a zero-shot

[3] $k = 1$ unless otherwise specified.

setup. Also, prior work has shown that designing cloze-style patterns and searching the answer space for NLI tasks (Schick and Schütze, 2021a; Webson and Pavlick, 2022) is difficult.

We also find that PARC-labeled noticeably outperforms PARC-unlabeled, indicating that the performance of self-prediction is limited by the capabilities of mBERT. More powerful MPLMs and better pattern designs might further improve the performance.

		En	**Af**	**Jv**	**Mn**	**My**	**Sw**	**Ta**	**Te**	**Tl**	**Ur**	**Uz**	**Avg**
MAJ		25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
Direct		52.5	41.8	27.4	42.5	32.2	31.3	31.5	33.0	31.6	46.9	44.8	36.3
UN	k=1	53.7	52.8	46.2	46.5	46.1	42.8	43.3	44.3	45.0	51.0	49.7	46.7
	k=3	55.8	53.6	46.2	47.1	48.2	44.9	44.5	46.3	47.1	52.6	51.0	48.1
	k=5	57.1	54.4	47.0	47.0	48.0	46.6	44.8	45.8	48.5	53.1	52.3	48.7
	k=10	57.5	55.3	46.3	46.4	47.6	45.6	44.1	46.7	47.7	53.0	51.4	48.4
	k=20	59.7	57.2	48.1	46.7	50.0	47.9	46.0	**48.9**	49.6	55.4	53.2	50.3
	k=30	**60.1**	**57.4**	**49.0**	**47.4**	**51.1**	**49.2**	**47.1**	48.7	**50.1**	**56.5**	**54.4**	**51.1**
LB	k=1	74.9	75.4	68.1	63.5	68.2	64.0	62.8	65.6	64.8	72.5	71.4	67.6
	k=3	77.1	77.1	69.6	65.6	71.1	67.6	65.6	68.4	65.9	74.6	74.4	70.0
	k=5	78.1	78.6	69.0	64.4	72.9	68.8	65.9	69.3	66.4	75.8	75.4	70.6
	k=10	78.7	79.4	70.5	67.0	72.9	68.3	66.6	70.7	67.2	76.6	75.9	71.5
	k=20	**79.0**	79.7	70.7	67.5	72.5	**70.0**	67.5	70.7	68.1	**77.4**	76.3	72.0
	k=30	79.0	**79.7**	**71.3**	**67.6**	72.8	69.9	**68.1**	**71.1**	**69.4**	77.2	**76.7**	**72.4**

Table 3.13: Results of topic categorization task on AG News dataset. k is the number of retrieved cross-lingual samples. MAJ is the majority baseline. Avg is the average accuracy across 10 LRLs. En is the HRL for retrieval.The BoR strategy is adopted.

To analyze the performance for every language in detail, we present the complete experimental results for the topic categorization task on AG News in Table 3.13. Here, we use the BoR method to take advantage of multiple retrieved HRL sentences. As expected, PARC outperforms the *Direct* baseline on all languages in both labeled and unlabeled settings.

However, it is worth noting that the sensitivity to cross-lingual retrieval differs from language to language. For some LRLs, e.g. Urdu (Ur) and Uzbek (Uz), PARC's improvement from cross-lingual retrieval is smaller. Others gain more, e.g. Javanese (Jv). Retrieving more samples increases the performance up to k=30 except for Telugu (Te) and Swahili (Sw), where the max is reached for k=20.

We now turn to the following two questions: 1) How does k affect the performance on tasks other than topic categorization? 2) Which LRLs profit most from our PARC method and which HRLs are best suited to retrieve prompts?

3.2.5 Analysis

3.2.5.1 Effect of k

We investigated how the performance changes as the number of retrieved HRL samples k increases. As shown in Figure 3.5, an abrupt accuracy increase can be seen in both labeled and unlabeled scenarios by concatenating the most similar cross-lingual sample. In labeled scenarios, the performance tends to increase up to k=20 and then levels off. This can be explained by

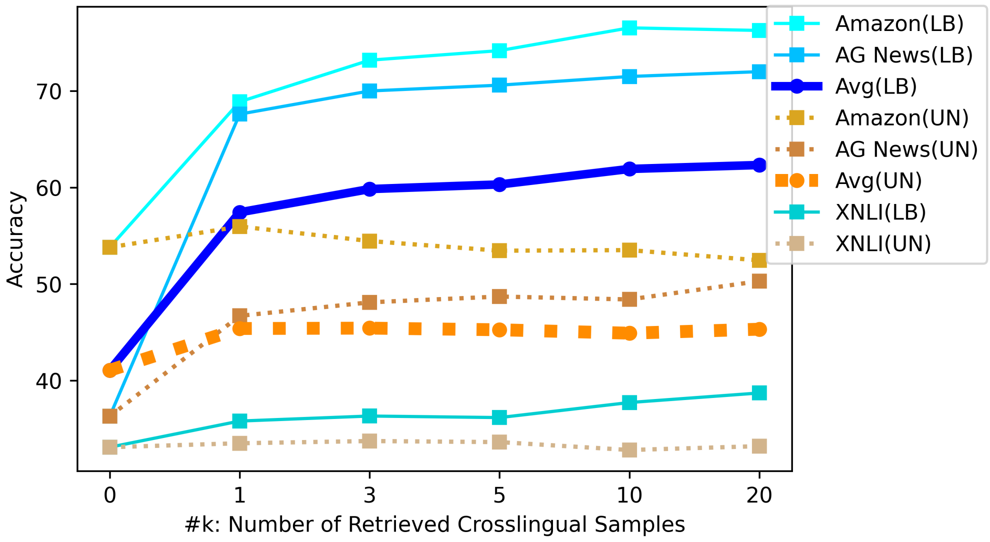


Figure 3.5: Accuracy on three tasks with different k in the labeled (LB) and unlabeled (UN) setup.

the fact that later retrieved samples are less similar to the input sample, so their contribution as prompts decreases. In unlabeled scenarios, there is no clear improvement beyond k=1 except for AGNews(UN), where the accuracy increases monotonically except for k=10. The performance of XNLI is less obviously influenced by the value of k than binary sentiment analysis and topic categorization. We assume that this could be attributed to the difficulty of the inference task. Unlike the other two single-sentence classification tasks, XNLI identifies the relationship between a pair of sentences. Transferring knowledge about sentence relationships is more complicated and requires more samples to learn, in contrast to the other two tasks where semantic information from similar cross-lingual sentences can be transferred directly.

3.2.5.2 Effect of Languages

Lauscher et al. (2020) pointed out that two linguistic factors exert crucial effects on cross-lingual transfer performance: (1) the size of the pretraining corpus for the target language and (2) the similarity between the source and target languages. In our study, we also consider a third factor: (3) the size of the pretraining corpus for the source language. In this section, we conduct a correlation analysis between PARC's cross-lingual transfer performance and the three language-related factors mentioned above. To achieve that, we first have to measure these factors properly. The size of the pretraining corpus can be easily measured by the log_2 value of the Wikipedia size in MB, as we mentioned in §3.2.3. Thus, the remaining problem is how to properly represent language similarity.

Measurement of Language Similarity Malaviya et al. (2017) and Littell et al. (2017) propose LANG2VEC from linguistic, typological, and phylogenetic perspectives. LANG2VEC employs

Lang	Language Similarity						Wiki Size
	SYN	PHO	INV	FAM	GEO	SIM	
Af	84.9	60.3	38.4	50.4	33.1	53.4	6
Jv	48.0	39.2	52.7	0.0	0.0	28.0	5
Mn	31.0	100.0	39.4	0.0	56.8	45.4	5
My	17.4	80.3	100.0	0.0	37.6	47.1	5
Ta	28.9	60.3	51.5	0.0	72.7	42.7	7
Te	36.0	56.2	31.3	0.0	45.2	33.7	7
Tl	35.0	70.5	26.7	0.0	38.8	34.2	6
Sw	27.0	87.0	62.1	0.0	57.2	46.6	5
Ur	50.2	72.0	47.1	12.6	62.5	48.9	7
Uz	39.8	75.6	24.1	0.0	73.7	42.6	6

Table 3.14: List of language features of the 10 LRLs that we evaluate.

different vectors to represent various types of linguistic features for different languages. Each language is encoded with 5 vectors corresponding to different linguistic features, including three typological features (syntax, phonology, and phonetic inventory), phylogenetic and geographical features. In typological vectors, each dimension represents a linguistic property. For example, one dimension of the syntax vector represents the word order feature SVO. If a language has an SVO order, then its syntax vector would have the value 1 on this dimension. Missing values in the typological vectors could have detrimental effects. Therefore, we replace them with values predicted from the k most similar typological vectors (Malaviya et al., 2017). The phylogenetic vector embodies the position of a language in the world language family tree (Harald et al., 2015), while the geographical vector contains the position information of languages w.r.t. their speakers.

Following prior work (Rama et al., 2020), we consider all 5 linguistic features when measuring the language similarity: syntax (SYN), phonology (PHO), phonological inventory (INV), language family (FAM), and geography (GEO). Given these different types of vectors, we calculate 5 cosine similarities for each pair of the source language (i) and target language (j) and average them to get the final language similarity $sim(i,j)$:

$$sim(i,j) = \frac{1}{|\mathcal{F}|}\sum_{f\in\mathcal{F}} s(\mathbf{v}_f(i), \mathbf{v}_f(j)) \tag{3.6}$$

where $\mathcal{F}$ is the set of features, $\mathbf{v}_f(i)$ and $\mathbf{v}_f(j)$ stand for the language vectors representing the feature f for i and j, and $s(\cdot)$ computes the min-max normalized cosine similarity of the two vectors. The detailed cosine similarities between English and 10 LRLs evaluated in our experiment are shown in Table 3.14. Language similarity refers to the similarity between each LRL and English. SIM score is computed by Eq. (3.6). WikiSize is the log value of the Wikipedia size in MB.

Unlabeled	Sim.		source size		target size	
	corr	p	corr	p	corr	p
Spearman	0.28	0.05	0.20	0.16*	0.31	0.03
Pearson	0.27	0.06*	0.22	0.12*	0.38	6e-03
labeled	Sim.		source size		target size	
	corr	p	corr	p	corr	p
Spearman	0.42	2e-03	0.08	0.54*	0.44	1e-03
Pearson	0.41	3e-03	-3e-4	1.00*	0.46	8e-4

Table 3.15: Correlations between Amazon review performance and three features. Sim.: language similarity between an LRL and an HRL; source (target) size: the log of the data size (MB) of the source (target). *: insignificant result with a p value larger than 0.05.

Correlation Analysis We conduct a correlation analysis between cross-lingual performance and the three language factors mentioned above: language similarity between the *source* (retrieved) and *target* (input) language, pretraining data size of the source language, and of the target language. We use the log value of Wikipedia size to represent the size of the pretraining corpus for target and source languages, and $sim(i, j)$ computed by Eq. (3.6) to represent the similarity between the source and target language. Four other HRLs – Chinese, German, Hindi, and Cebuano – are selected as source languages in addition to English. We measure the cross-lingual performance of PARC on the Amazon product review task in both the labeled and the unlabeled settings. Table 3.17 shows the detailed data used for correlation analysis of language similarity, high- and low-resource language pretraining data size with cross-lingual performance in the unlabeled setting as well as the labeled setting.

Table 3.15 shows the outcome of the correlation analysis. We observe a significant positive correlation between cross-lingual performance and language similarity as well as target language pretraining data size, in both the labeled and the unlabeled setting. The correlation between performance and source language size is not significant. Figure 3.6 visualizes the correlations and further clarifies the findings by selecting 4 source languages and 4 target languages and showing the cross-lingual performance and similarity between them.

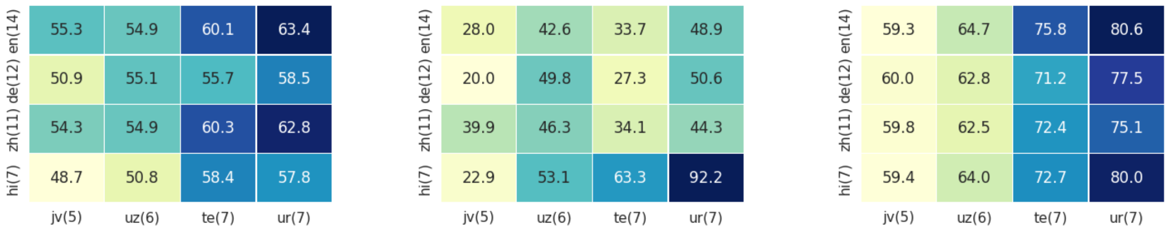
3.2.5.3 Robustness

In this section, we test the robustness of the PARC method w.r.t. other cross-lingual retrievers and MPLMs as well as unseen languages.

Retriever and MPLM Apart from the multilingual sentence transformer based on XLM-R ("paraphrase") used in our previous experiments, we explore several other types of cross-lingual retrievers: a "pooling" retriever, which averages the last hidden states of the MPLM and computes the cosine similarity between these pooled sentence representations; a "distiluse" retriever, a sentence transformer based on multilingual distilBERT (Sanh et al., 2019); and the "LaBSE"

		Amazon	**AGNews**	**XNLI**	**Avg.**
Direct		53.8	36.2	33.1	41.0
UN	mBERT+pooling	53.1	36.9	33.6	41.2
	mBERT+distiluse	54.7	38.4	34.0	42.3
	mBERT+paraphrase	59.6	46.7	33.7	46.7
	XLM-R+paraphrase	**70.1**	**57.4**	34.7	**54.1**
	mBERT+LaBSE	59.4	43.8	**35.1**	46.1
LB	mBERT+pooling	53.6	58.0	33.8	48.5
	mBERT+distiluse	62.8	63.8	34.6	53.7
	mBERT+paraphrase	72.9	67.6	36.8	59.1
	XLM-R+paraphrase	**73.0**	76.0	35.7	61.6
	mBERT+LaBSE	72.2	**80.0**	**37.5**	**63.2**

Table 3.16: Accuracy with different models used in our approach. pooling: cosine similarity of the last hidden states from the MPLM; distiluse: *distiluse-base-multilingual-cased-v2*, sentence transformer of multilingual distilBERT; paraphrase: *paraphrase-multilingual-mpnet-base-v2*, sentence transformer of XLM-R. UN: unlabeled setup; LB: labeled setup.



(a) Zero-Shot Performance (Unlabeled)   (b) Language Similarity   (c) Zero-Shot Performance (labeled)

Figure 3.6: Visualization of the correlation between zero-shot performance and language similarity, pretraining data size of source and target language. On the X(Y)-axis are target(source) languages with an increasing order of pretraining data size from left(bottom) to right(top). (a) and (c) show the zero-shot performance with PARC-unlabeled and PARC-labeled on the Amazon review task, respectively. (b) shows the language similarity of each pair.

retriever (Feng et al., 2022), a BERT-based model trained for sentence embedding for 109 languages. As an alternative to mBERT, we also investigate the performance of XLM-R, which has the same architecture as mBERT but is more powerful. We follow the setup described in §3.2.3.

Results are shown in Table 3.16. We can find that even the worst combination—*mBERT+pooling*—outperforms the *Direct* baseline on average under both labeled and unlabeled settings. If the retriever is replaced by a slightly more powerful one, such as the combination *mBERT+distiluse*, higher accuracies in the unlabeled and labeled settings are achieved, suggesting that our proposed method PARC is robust w.r.t. other cross-lingual retrievers. In the result of *XLM-*

	Performance		**Language Similarity**						**WikiSize**	
	Unlabeled	labeled	SYN	PHO	INV	FAM	GEO	SIM	source	target
en-af	79.2	62.0	84.9	60.3	38.4	50.4	33.1	53.4	14	6
en-ur	80.6	63.4	50.2	72.0	47.1	12.6	62.5	48.9	14	7
en-sw	49.9	51.0	27.0	87.0	62.1	0.0	57.2	46.6	14	5
en-te	75.8	60.1	36.0	56.2	31.3	0.0	45.2	33.7	14	7
en-ta	75.4	60.2	28.9	60.3	51.5	0.0	72.7	42.7	14	7
en-mn	74.9	62.9	31.0	100.0	39.4	0.0	56.8	45.4	14	5
en-uz	64.7	54.9	39.8	75.6	24.1	0.0	73.7	42.6	14	6
en-my	73.8	60.3	17.4	80.3	100.0	0.0	37.6	47.1	14	5
en-jv	59.3	55.3	48.0	39.2	52.7	0.0	0.0	28.0	14	5
en-tl	55.4	53.5	35.0	70.5	26.7	0.0	38.8	34.2	14	6
de-af	71.6	56.5	87.1	33.1	90.3	77.2	43.1	66.2	12	6
de-ur	77.5	58.5	50.7	68.3	45.8	15.4	72.6	50.6	12	7
de-sw	50.6	48.9	29.5	33.1	36.2	0.0	66.7	33.1	12	5
de-te	71.2	55.7	45.6	29.4	5.2	0.0	56.5	27.3	12	7
de-ta	76.3	57.6	43.0	56.7	48.7	0.0	81.3	45.9	12	7
de-mn	74.7	59.1	44.4	68.3	42.8	0.0	61.8	43.4	12	5
de-uz	62.8	55.1	48.3	91.9	27.8	0.0	81.1	49.8	12	6
de-my	72.0	59.3	31.3	29.9	63.9	0.0	47.5	34.5	12	5
de-jv	60.0	50.9	41.5	14.4	32.5	0.0	10.3	19.8	12	5
de-tl	54.5	52.1	48.1	42.1	0.0	0.0	50.8	28.2	12	6
zh-af	70.4	58.6	53.9	9.5	25.2	0.0	12.1	20.1	11	6
zh-ur	75.1	62.8	59.0	43.5	36.3	0.0	82.6	44.3	11	7
zh-sw	53.9	51.5	5.7	33.1	27.0	0.0	27.6	18.7	11	5
zh-te	72.4	60.3	49.9	29.4	4.5	0.0	86.7	34.1	11	7
zh-ta	73.0	61.8	19.0	56.7	16.8	0.0	40.5	26.6	11	7
zh-mn	71.6	60.4	56.5	43.5	8.7	0.0	99.0	41.5	11	5
zh-uz	62.5	54.9	49.0	69.3	26.2	0.0	87.2	46.3	11	6
zh-my	69.6	59.3	42.5	71.8	32.7	37.8	95.7	56.1	11	5
zh-jv	59.8	54.3	41.1	42.1	31.4	0.0	85.1	39.9	11	5
zh-tl	54.7	52.4	44.7	14.4	6.9	0.0	83.4	29.9	11	6
hi-af	78.2	59.0	55.4	50.1	30.8	14.3	52.3	40.6	7	6
hi-ur	80.0	57.8	100.0	88.1	73.0	100.0	99.9	92.2	7	7
hi-sw	50.7	50.5	27.4	24.6	24.9	0.0	66.9	28.8	7	5
hi-te	72.7	58.4	74.7	74.4	67.2	0.0	100.0	63.3	7	7
hi-ta	74.2	57.0	48.9	50.1	36.8	0.0	75.8	42.3	7	7
hi-mn	74.6	57.7	57.9	61.3	31.2	0.0	89.4	48.0	7	5
hi-uz	64.0	50.8	57.8	64.8	45.6	0.0	97.2	53.1	7	6
hi-my	74.3	58.7	36.7	46.7	37.5	0.0	97.6	43.7	7	5
hi-jv	59.4	48.7	21.2	0.0	13.6	0.0	79.6	22.9	7	5
hi-tl	56.6	52.9	73.1	59.8	41.3	0.0	98.2	54.5	7	6
ceb-af	63.9	58.1	42.4	44.1	52.5	0.0	8.9	29.6	11	6
ceb-ur	68.7	57.1	29.3	84.3	22.5	0.0	62.9	39.8	11	7
ceb-sw	53.4	49.2	33.0	16.1	76.3	0.0	12.0	27.5	11	5
ceb-te	69.3	59.0	4.8	98.6	17.9	0.0	75.9	39.4	11	7
ceb-ta	66.3	55.8	22.4	72.1	63.0	0.0	16.6	34.8	11	7
ceb-mn	65.9	59.7	16.5	55.0	37.6	0.0	79.3	37.7	11	5
ceb-uz	56.2	52.6	26.2	61.3	17.9	0.0	60.6	33.2	11	6
ceb-my	64.8	56.3	3.0	43.5	57.7	0.0	88.1	38.4	11	5
ceb-jv	57.1	51.2	60.2	17.1	70.0	54.8	97.6	59.9	11	5
ceb-tl	53.0	56.2	0.0	82.7	50.0	0.0	76.2	41.8	11	6

Table 3.17: Detailed data of 50 source-target language pairs used for correlation analysis of language similarity, source and target language pretraining data size with cross-lingual performance in unlabeled and labeled setups. Task performance is measured on the Amazon review task with $k = 1$.

		Ig	Sn	Mt	Co	Sm	St	Haw	Zu	Ny	Avg.
Direct		30.3	32.1	29.8	32.6	30.4	30.4	27.1	34.4	29.8	30.8
LB	k=1	56.5	59.7	63.9	75.0	52.0	53.5	49.9	58.0	54.9	58.1
	k=3	58.1	61.4	65.2	78.2	54.1	55.5	49.7	58.5	57.0	59.7
	k=5	**58.8**	**61.6**	**65.9**	**79.8**	**55.4**	**56.8**	**51.4**	**58.8**	**58.0**	**60.7**
UN	k=1	36.6	37.3	39.1	42.6	34.4	36.3	31.6	35.6	35.3	36.5
	k=3	34.8	36.2	37.6	40.6	33.9	33.7	31.0	34.3	32.9	35.0
	k=5	34.8	35.3	37.2	40.4	34.1	34.2	30.6	34.0	32.0	34.7

Table 3.18: Results of several unseen languages on a topic categorization task (AG News dataset). Ig - Igbo, Sn - Shona, Mt - Maltese, Co - Corsican, Sm - Samoan, St - Sesotho, Haw - Hawaiian, Zu - Zulu, Ny - Chichewa.

		p1		p2		p3		p4		Avg	
		en	te	en	te	en	te	en	te	en	te
Finetune	Direct	84	76	83	70	86	67	85	73	85	74
	PARC-UN	84 –	65↓	85↑	62↓	83↓	60↓	82↓	64↓	84↓	67↓
	PARC-LB	83↓	64↓	83 –	64↓	83↓	64↓	82↓	70↓	83↓	69↓
w/o Finetune	Direct	54	53	59	54	54	50	53	51	55	52
	PARC-UN	59↑	55↑	55↓	58↑	52↓	52↑	53 –	52↑	55 –	54↑
	PARC-LB	**90↑**	**82↑**	**90↑**	**82↑**	**90↑**	**82↑**	**90↑**	**82↑**	**90↑**	**82↑**

Table 3.19: Result of English and Telugu on Amazon review task using MPLMs with and without fine-tuning on English train set. UN: Unlabeled, LB: labeled. p_i represents different prompt patterns.

R+paraphrase, the obviously better performance of XLM-R in the unlabeled setup shows that a stronger MPLM can noticeably improve the self-prediction. We expect that an even better performance could be obtained by applying our proposed PARC approach to larger and/or more powerful MPLMs such as InfoXLM (Chi et al., 2021).

Unseen Languages Our previous experiments show that the LRLs pretrained by MPLMs can benefit well from PARC. However, popular MPLMs are pretrained only on approximately. 100 languages, accounting for a tiny part of all languages in the world (∼100/7000). We wonder if our proposed method could potentially benefit a wider range of LRLs, so we apply PARC to

Amazon Review

Case #963
Input:
အဝတ်လျှော်အများအပြားဝန်နှင့်အတူအသုံးပြုခဲ့ကြသည်။ထည်အတွက်နူးညံ့သိမ်မွေ့ခြင်း
နှင့်ငါ့အသားအရေကိုနူးညံ့သိမ်မွေ့ပုံရသည်။

(Used with several loads of laundry. Gentle on the fabric and gentle on my skin.) pos
Retrieved:
R1: Hard to wash. The fur on top gets all over the sides in the wash. :/ pos
R2: Very nice and thick high quality towels. pos
R3: Smelled really bad mold! I had to wash them before use. neg
Predictions: No retrieval - neg, **k=1 -** neg, **k=3 -** pos

Table 3.20: A PARC pipeline example for the Amazon review task in the labeled setting.

several unseen LRLs, i.e., languages not included in the pretrained corpora of the MPLM. We conduct experiments on a topic categorization task for nine unseen languages. The results in Table 3.18 show that PARC is also effective for unseen LRLs. It can be observed from the result that PARC is also effective for unseen LRL languages.

3.2.5.4 Zero-shot Setting

Different from the cross-lingual transfer paradigm where an MPLM is first finetuned on annotated training data of one language, and then directly applied to the test data of other languages for inference, our proposed approach is employed in the zero-shot setting for LRLs, i.e., the model parameters are not adjusted by fine-tuning with HRL data. Table 3.19 shows results from a preliminary experiment where our PARC method, combined with a finetuned MPLM, fails to outperform the Direct baseline. When using finetuned MPLM to evaluate with PARC, we do not see sufficient performance improvement. However, without fine-tuning, PARC performs better in both unlabeled and labeled setups, and PARC-LB without fine-tuning also outperforms it with fine-tuning.

3.2.5.5 Qualitative Analysis

Table 3.20 shows the results of the PARC pipeline for an example from the Amazon review task. The review in Telugu is positive, but the class predicted without cross-lingual context is negative. The prediction stays the same when a single positive English sample is added as prompt context. When two more English samples are added, the prediction becomes correct.

This case indicates that the retrieved cross-lingual samples help the MPLM make a correct decision. Furthermore, more similar HRL samples could rectify the deviation. More cases are

Amazon Review
Case 1 #37 **Input:** ငါ့ဆံပင်ပေါ်မှာအလွန်ခြောက်သွေ့။ (Very dry on my hair.) neg **Retrieved**: **R1**: It’s a little bit too greasy in my opinion. Doesn’t really seem to soak into the hair very well. pos **R2**: The tiniest amount leaves my hair stringy and oily. neg **R3**: could smell this stuff all day but I don’t feel like it moisturizes my skin enough, and my skin isn’t overly dry to begin with. pos **Predictions: No retrieval -** pos, **k=1 -** neg, **k=3 -** neg
Case 2 #963 **Input:** အဝတ်လျှော်အများအပြားဝန်နှင့်အတူအသုံးပြုခဲ့ကြသည်။ထည်အတွက်နူးညံ့သိမ်မွေ့ခြင်း နှင့်ငါ့အသားအရေကိုနူးညံ့သိမ်မွေ့ပုံရသည်။ (Used with several loads of laundry. Gentle on the fabric and gentle on my skin.) pos **Retrieved**: **R1**: Hard to wash. The fur on top gets all over the sides in the wash. :/ pos **R2**: Very nice and thick high quality towels. pos **R3**: Smelled really bad mold! I had to wash them before use. neg **Predictions: No retrieval -** neg, **k=1 -** neg, **k=3 -** pos

Table 3.21: PARC examples for Amazon Review task.

shown in Table 3.21 and Table 3.22. Table 3.21 shows two examples from the Amazon Review task. We compare the predictions for three scenarios: no retrieval information (i.e., Direct baseline, see §3.2.3.2), one retrieved sample, and three retrieved samples. Similarly, Table 3.22 shows the same comparison on the AG News task.

3.2.6 Sum-Up

In this section, we propose PARC, a pipeline that augments prompts for zero-shot learning on low-resource languages by retrieving semantically similar cross-lingual sentences from HRL corpora. We test PARC on three classification tasks with parallel test sets across 10 LRLs, and it performs better than the baselines in both unlabeled and labeled settings. Increasing the number of retrieved prompts improves performance at first, but deteriorates it after a certain point. A robustness study shows that PARC also performs well with other cross-lingual retrievers

AG News

Case 1 #1939
Input:
ပန်းပွင့်ပါဝါသည်ပန်းများကိုအသံချဲ့စက်များသို့လှည့်လာသည့်နည်းလမ်းဖြင့်ဂျပန်ကုမ္ပဏီ
တစ်ခုပေါ်လာသည်။

(Flower Power A Japanese company has come up with a way to turn flowers into amplifiers.) Tech
Retrieved:
R1: Japanese firms step up spending Japanese firms continue to spend on new equipment and production plants, a survey finds, underlining a continuing recovery in the world's second-largest economy. Business
R2: IBM, Honda deliver in-car speech-recognition navigation system IBM and Honda have jointly developed a hands-free and natural sounding in-vehicle speech-recognition system that will be offered as standard equipment on the 2005 Acura RL Tech
R3: Scientists Make Phone That Turns Into a Sunflower (Reuters) Reuters - Scientists said on Monday they have come up with a cell phone cover that will grow into a sunflower when thrown away. Tech
Predictions: No retrieval - World, **k=1 -** Tech, **k=3 -** Tech

Case 2 #1302
Input:
လျှပ်တပြက်အတွင်း ရုပ်ရှင်များ- Netflix နှင့် TiVo တို့သည် ဒေါင်းလုဒ်များကို Bee
Staff Writer ဆွေးနွေးကြသည်။ TiVo Inc ၏ပိုင်ရှင်များကိုခွင့်ပြုမည့် Silicon Valley
မဟာမိတ်အသစ်၏အသံများကြားတွင်နည်းပညာမြင့်မြေပြင်သည်ခြေလျင်အောက်သို့
ပြောင်းလာသည်။

(Movies in a Snap: Netflix and TiVo Discuss Downloads Bee Staff Writer. The high-tech terrain is shifting underfoot amid rumblings of a new Silicon Valley alliance that would allow the owners of TiVo Inc.) Business
Retrieved:
R1: NETFLIX, TIVO HOOKUP CLOSE Netflix and TiVo are in late-stage talks on a partnership that would let subscribers use the Internet to download Netflix movies directly into their TiVo box, The Post has learned. Business
R2: TiVo and NetFlix: Picture-Perfect Duo? With TiVo (TIVO) and NetFlix (NFLX) finally announcing a long-rumored partnership to launch a video-on-demand service sometime next year, investors smiled on the deal that will keep the two popular, but under-fire, innovators ahead of competitors. Tech
R3: New Treo and more unveiled at CTIA CTIA stands for the Cellular Telecommunications and Internet Association. Each year they host two shows for the industry. This week is their fall Wireless IT and Entertainment expo in San Francisco. Business
Predictions: No retrieval - World, **k=1 -** Tech, **k=3 -** Business

Table 3.22: PARC examples for AG News task

or MPLMs, suggesting potential applications of PARC to a wider scope of scenarios. The PARC pipeline proposed in this work is designed to improve the cross-lingual transfer performance for low-resource languages in a zero-shot setting. We tested our method on different LRLs contained in MPLMs and also investigated its effectiveness for several unseen languages. These are not included in pretraining corpora of the MPLM but use a seen script and share some subwords with the seen languages. However, our proposed method is not applicable to unseen languages with new scripts, which restricts its extension towards a wider range of languages. Besides, PARC is a retrieval-based method. More time and computational resources are required in the cross-lingual retrieval phase. Therefore, it is computationally less efficient to use PARC for inference.

3.3 Decomposed Prompting for Multilingual Linguistic Structure Knowledge Evaluation

This section corresponds to the following work:

> **Ercong Nie**, Shuzhou Yuan, Bolei Ma, Helmut Schmid, Michael Färber, Frauke Kreuter, Hinrich Schütze. 2024. Decomposed Prompting: Unveiling Multilingual Linguistic Structure Knowledge in English-Centric Large Language Models. In Findings of the Association for Computational Linguistics: IJCNLP-AACL 2025, Mumbai, India. Association for Computational Linguistics.

Declaration of Co-Authorship. I conceived the idea of using decomposed prompting for multilingual sequence labeling tasks and applied it to English-centric Large Language Models (LLMs) for linguistic structure knowledge evaluation. I conducted all the experiments and implemented the baselines. I finished most of the first draft. Shuzhou Yuan and Bolei Ma contributed by drawing the figures and graphs, discussing the idea, and writing part of the paper. The other authors are supervisors who supervised the project process and provided valuable feedback throughout the project.

Summary of This Section

Despite the predominance of English in their training data, English-centric Large Language Models (LLMs) like GPT-3 and LLaMA display a remarkable ability to perform multilingual tasks, raising questions about the depth and nature of their cross-lingual capabilities. This section introduces the *decomposed prompting* approach to probe the linguistic structure understanding of these LLMs in sequence labeling tasks. Diverging from the single text-to-text prompt, our method generates for each token of the input sentence an individual prompt that asks for its linguistic label. We assess our method on the Universal Dependencies part-of-speech tagging dataset for 38 languages, utilizing both English-centric and multilingual LLMs. Our findings show that *decomposed prompting* surpasses the *iterative prompting* baseline in efficacy and efficiency under zero- and few-shot settings. Further analysis reveals the influence of evaluation methods and the use of instructions in prompts. Our multilingual investigation shows that English-centric language models perform better on average than multilingual models. Our study offers insights into the multilingual transferability of English-centric LLMs, contributing to the understanding of their multilingual linguistic knowledge.

3.3.1 Motivation and Research Question

Current Large Language Models (LLMs), such as GPT-3, GPT-4, PaLM, and LLaMA (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023a), have demonstrated remarkable capabilities in in-context learning, also known as prompting, across a broad spectrum of language understanding and generation tasks (Zhao et al., 2023; Zhang et al., 2023c; Ziyu et al., 2023). These models are predominantly trained on massive amounts of English text data, with some limited exposure to other languages. For instance, LLaMA2's pretraining corpus comprises over 89% English content (Touvron et al., 2023b). Yet, these English-centric LLMs [4] still exhibit effective performance in multilingual evaluations (Lai et al., 2023a). In the previous section, we introduced the multilingual prompting scenario designed for zero-shot transfer. In this section, we apply this method to English-centric LLMs, where the model executes tasks by directly generating outputs based on a task description and/or a few examples provided in a pivot language (typically English), along with input in a different target language (Ahuja et al., 2023). However, the extent and nature of their cross-lingual capabilities remain underexplored (Ye et al., 2023). This raises a critical question: *Does the multilinguality of these models stem from a deep, generalizable multilingual linguistic understanding, or merely from the superficial alignment of lexical patterns across languages?*

Given the demonstrated proficiency of English-centric LLMs in multilingual tasks that demand profound language understanding (Deng et al., 2023; Wang et al., 2023d), we hypothesize that these models harbor substantial multilingual knowledge. This knowledge, particularly relating to linguistic structure, is commonly conceptualized through sequence tagging tasks (Jurafsky and Martin, 2000). However, the current prompting strategies designed for sequence labeling in LLMs are not well-suited for testing. For instance, behavioral probing methods (Belinkov et al.,

[4] In our work, we regard a model pretrained primarily on English text as English-centric.

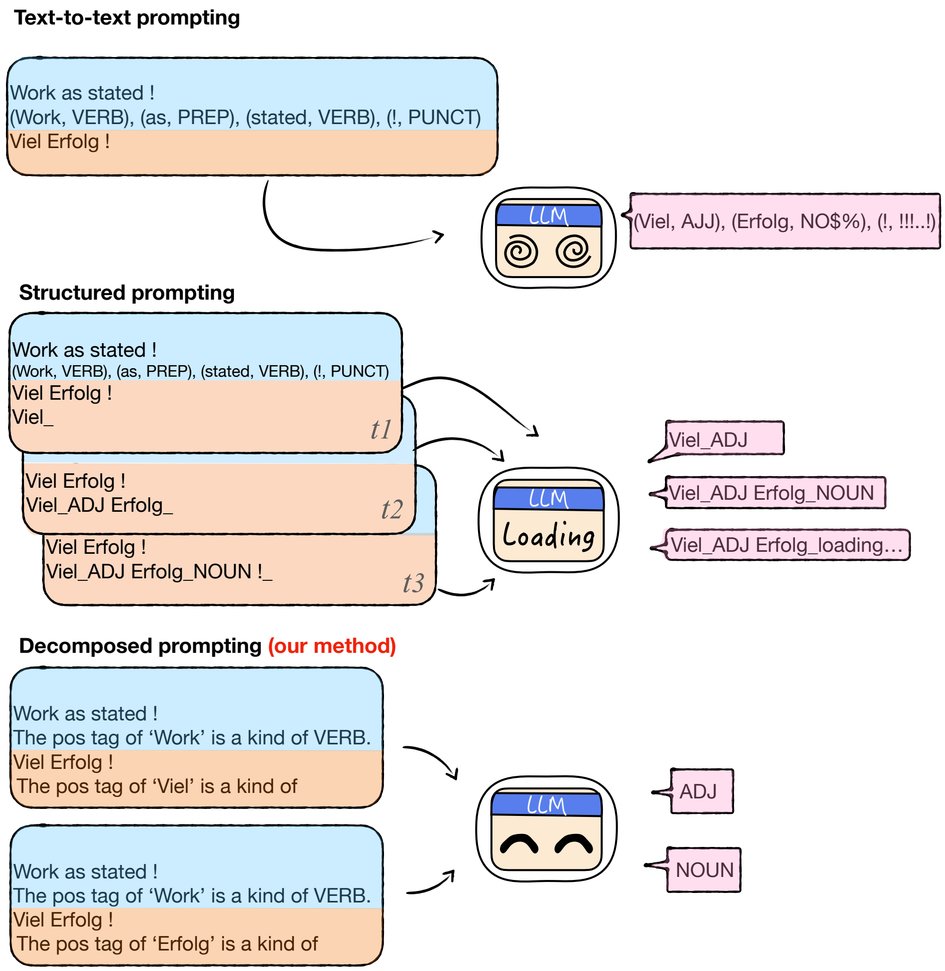

Text-to-text prompting
Work as stated !
(Work, VERB), (as, PREP), (stated, VERB), (!, PUNCT)
Viel Erfolg !
LLM
(Viel, AJJ), (Erfolg, NO$%), (!, !!!..!)
Structured prompting
Work as stated !
(Work, VERB), (as, PREP), (stated, VERB), (!, PUNCT)
Viel Erfolg !
Viel_
t1
Viel Erfolg !
Viel_ADJ Erfolg_
t2
Viel Erfolg !
Viel_ADJ Erfolg_NOUN !_
t3
LLM
Loading
Viel_ADJ
Viel_ADJ Erfolg_NOUN
Viel_ADJ Erfolg_loading…
Decomposed prompting (our method)
Work as stated !
The pos tag of 'Work' is a kind of VERB.
Viel Erfolg !
The pos tag of 'Viel' is a kind of
Work as stated !
The pos tag of 'Work' is a kind of VERB.
Viel Erfolg !
The pos tag of 'Erfolg' is a kind of
LLM
ADJ
NOUN

Figure 3.7: Comparison of different prompting methods for sequence labeling.

2020), aimed at measuring knowledge stored in language models, struggle to adapt to tasks predicting more complex structures. Additionally, text-to-text prompting methods (Asai et al., 2024), which rely on a predefined output template, face challenges in maintaining control over the output format. In response to these challenges, a suitable *iterative prompting* strategy for structured prediction has been introduced, addressing the aforementioned limitations (Blevins et al., 2023). Despite its advantages, this method presents its own challenges, such as longer processing times due to its iterative inference strategy.

To overcome the challenges identified in probing the multilingual knowledge of linguistic structure in LLMs, we introduce the *decomposed prompting* strategy. We adopt the idea of decomposing a single prompt into multiple prompts to the in-context learning paradigm, aiming to probe English-centric LLMs for their understanding of token-level linguistic structure framed as sequence labeling tasks. As shown in Figure 3.7, instead of employing a single text-to-text prompt for labeling an entire sequence in one step, our method decomposes this process into multiple discrete prompts. More precisely, we first split the input sentence into tokens. Subsequently, we generate an individual prompt for each token that inquires about its linguistic label.

We evaluate our approach on the Universal Dependency (UD) part-of-speech (POS) tagging dataset (Nivre et al., 2020) covering 38 languages with 3 English-centric LLMs and 2 multilingual LLMs. Our approach outperforms the iterative prompting baseline in both zero- and few-shot settings in terms of accuracy and efficiency. We investigate the nuanced impact of evaluation methods and the usage of task instructions within prompts on the performance of decomposed prompting, followed by an empirical comparative study of decomposed and iterative prompting. Moreover, our analysis of the multilingual efficacy of English-centric LLMs yields valuable insights into the transferability of linguistic knowledge via multilingual prompting.

3.3.2 Decomposed Prompting for LLMs

In this study, we introduce a novel approach for conducting sequence labeling with LLMs through in-context learning, termed *decomposed prompting*.

Intuition This method draws inspiration from the step-by-step thinking process humans employ when annotating linguistic features within a sentence. Typically, humans approach such tasks incrementally, addressing each token individually. Mirroring this intuitive strategy, our method first decomposes an input sentence into tokens. Subsequently, we generate a distinct prompt for each token, thereby transforming the sequence labeling task into a series of focused, manageable prompts. Figure 3.8 illustrates the generation of sequence labeling prompts for the German sentence *"Viel Erfolg!"* via *decomposed prompting*.

Problem Formulation Given a test sequence set $\mathcal{X}_{test}$, a label set L, and an LLM M, we approach the task of sequence labeling as follows: for an input sequence $X \in \mathcal{X}_{test}$ of length n, $X = x_1, \cdots, x_n$, the model M is expected to produce a corresponding sequence of labels $\hat{Y} = \hat{y}_1, \cdots, \hat{y}_n$, where each label $\hat{y}_i \in L$ is associated with the linguistic feature of the token x_i.

In decomposed prompting, we design a prompt template function $T(\cdot, \cdot)$ which generates a specific prompt for each token. T takes the input sequence X and an individual token x_i as arguments and returns a prompt for predicting the label of the token. The true label y_i can be optionally included as an argument to T; if included, T will generate a demonstration. An example of such a template function is illustrated as follows.

> $T(X, x_i) =$ "Sentence: X. In the sentence, the part-of-speech tag of 'x_i' is a kind of"
> $T(X, x_i, y_i) =$ "Sentence: X. In the sentence, the part-of-speech tag of 'x_i' is a kind of y_i."

$C = c_1, \cdots, c_m$ is a sample from the training set. In the few-shot learning scenario, k examples in the tuple format (C_j, c_j, l_j) are given along with the input sequence X, where c_j is a token in C_j, and $l_j \in L$ is the label for c_j. The demonstration D of an input sequence X is formulated as:

$$D = I \circ T(C_1, c_1, l_1) \circ \cdots \circ T(C_k, c_k, l_k) \tag{3.7}$$

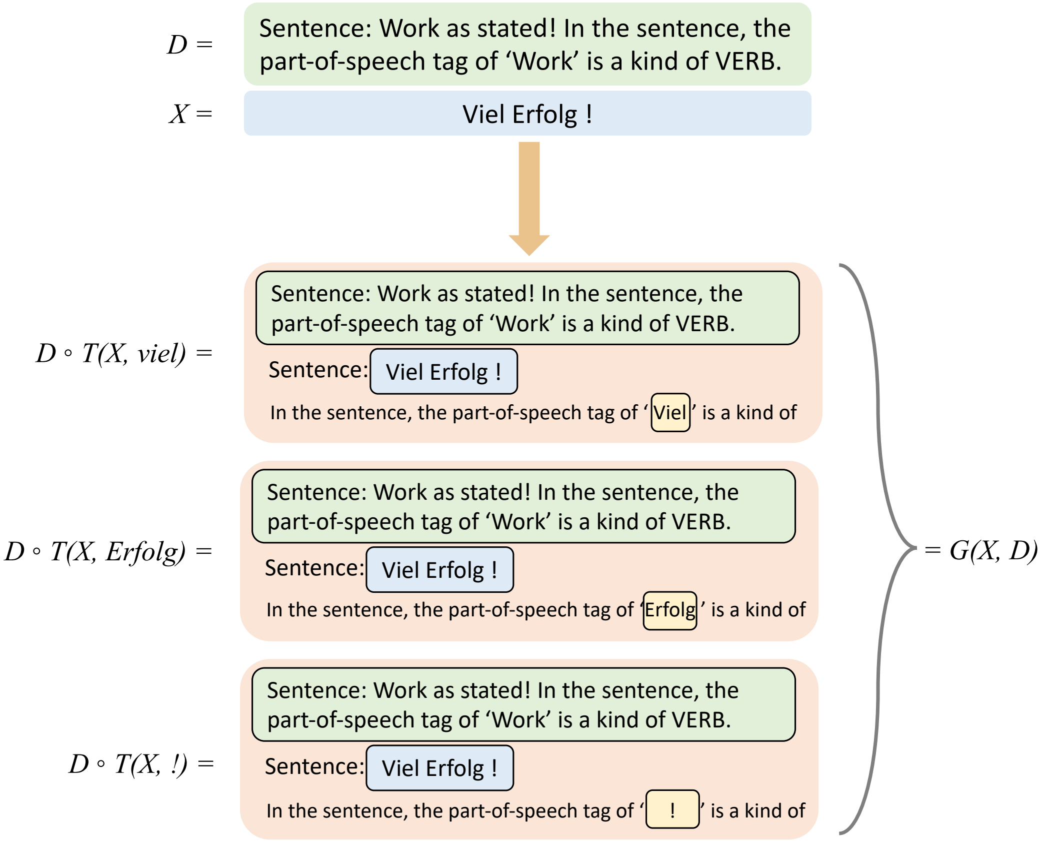

D =
Sentence: Work as stated! In the sentence, the part-of-speech tag of 'Work' is a kind of VERB.
X =
Viel Erfolg !
D ∘ T(X, viel) =
Sentence: Work as stated! In the sentence, the part-of-speech tag of 'Work' is a kind of VERB.
Sentence: Viel Erfolg !
In the sentence, the part-of-speech tag of 'Viel' is a kind of
D ∘ T(X, Erfolg) =
Sentence: Work as stated! In the sentence, the part-of-speech tag of 'Work' is a kind of VERB.
Sentence: Viel Erfolg !
In the sentence, the part-of-speech tag of 'Erfolg' is a kind of
D ∘ T(X, !) =
Sentence: Work as stated! In the sentence, the part-of-speech tag of 'Work' is a kind of VERB.
Sentence: Viel Erfolg !
In the sentence, the part-of-speech tag of ' ! ' is a kind of
= G(X, D)

Figure 3.8: An example of how *decomposed prompting* is implemented for sequence labeling.

where I denotes an optional instruction in natural language, $\circ$ denotes the string concatenation operation. Finally, we use a prompt generator function $G(\cdot, \cdot)$ to create the set of decomposed prompts for an input sequence X:

$$G(X, D) = \{D \circ\ T(X, x_1), \cdots, D \circ\ T(X, x_m)\} \tag{3.8}$$

The label $\hat{y}_i$ of token x_i is predicted as follows:

$$\hat{y}_i = \operatorname*{argmax}_{y \in L} P_M(l | D \circ\ T(X, x_i)) \tag{3.9}$$

For each possible label y, we obtain the probability that the model predicts this label as the next token and select the most likely label as the predicted label.

3.3.3 Experimental Setup

3.3.3.1 Datasets and Languages

In our study, we focus on evaluating the multilingual linguistic structure knowledge of English-centric models through multilingual part-of-speech tagging tasks, employing our proposed decomposed prompting method. We utilize a subset of the Universal Dependency treebanks (UD-POS) (Nivre et al., 2020) for this purpose. The UDPOS dataset adopts a universal POS tag set consisting of 17 tags. Figure 3.9 shows the pos tag set in UD. We also use the text in the box as the task instruction in our experiments.

```
POS tag set:  ADJ ADP ADV AUX CCONJ DET INTJ NOUN NUM PART PRON
PROPN PUNCT SCONJ SYM VERB X
```

Figure 3.9: UD POS tag set.

Our chosen subset, derived from the XTREME multilingual benchmark (Hu et al., 2020b), comprises 38 languages from diverse language families, as Figure 3.10 shows. If the test set of a language contains more than 200 sentences, we randomly sample 200 instances for the evaluation due to computational constraints.

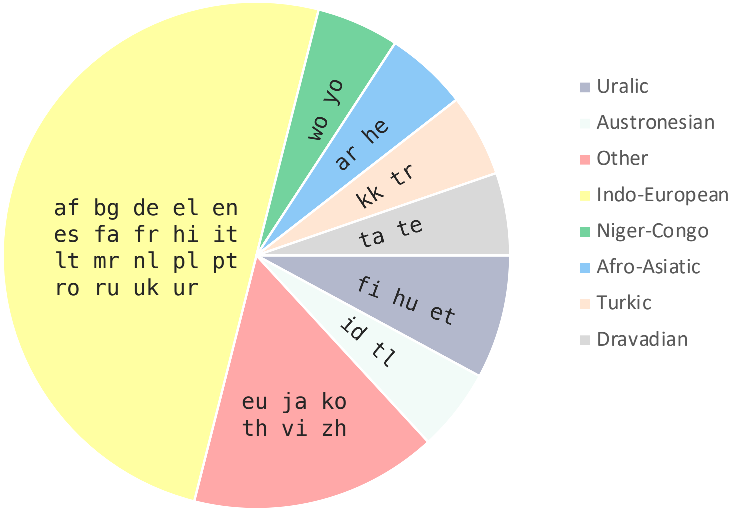


Figure 3.10: Distribution of languages by language family in the dataset.

3.3.3.2 Models

We select a diverse list of LLMs, including three English-centric LLMs and two multilingual LLMs in order to investigate the differences in multilingual understanding across LLMs with varying degrees of multilinguality and base capabilities. All LLMs in this experiment are instruction-tuned versions accessible through the HuggingFace framework (Wolf et al., 2020).

English-centric LLMs **LLaMA2** represents an advanced iteration of the LLaMA foundation models developed by Meta AI (Touvron et al., 2023a,b), trained on publicly available corpora predominantly in English. Compared to its predecessor, LLaMA2 benefits from an enhanced data cleaning process, expanded language coverage, and the implementation of more efficient grouped-query attention (Ainslie et al., 2023). We consider LLaMA2 models with 7B and 13B parameters in our experiments. **Mistral 7B** (Jiang et al., 2023) enhances the LLaMA models in terms of both performance and inference efficiency, achieved through meticulous engineering in language model design and training. For our experiments, we utilize the instruction-tuned version of Mistral 7B, which has been fine-tuned on the OpenHermes 2.5 dataset [5].

Multilingual LLMs **BLOOMZ** (Muennighoff et al., 2023) is a multi-task fine-tuned variant of the BLOOM model (Workshop et al., 2022), which is trained on 46 languages. We employ its 7B version in our experiment. **mTk-Instruct** (Wang et al., 2022c) is a multilingual encoder-decoder model, fine-tuned on instruction-following datasets. The datasets features instructions generated by GPT-4 (Achiam et al., 2023; Peng et al., 2023). mTk-Instruct is built upon the mT5 model (Xue et al., 2021), which is pretrained on corpora of over 100 languages. It comprises approximately 13 billion parameters.

3.3.3.3 Baselines and Settings

Iterative Prompting (*Iter*) Blevins et al. (2023) introduced a structured prompting approach that *iteratively* labels an entire sentence by appending each predicted label to the context along with the subsequent word(see Figure 3.7). This method is employed as a strong baseline in our study.

Decomposed Prompting (*Decom*) To evaluate our proposed approach, we employ the prompt template outlined in §3.3.2 to decompose the entire sequence into a set of individual prompts for prediction. In our experiments, we use the 17 POS tags themselves as the label words, i.e., we expect the model to directly predict a tag from the tagset shown in Figure 3.9 by selecting the tag with the highest logit.

Zero- and Few-Shot Prompting We devised two experimental scenarios for multilingual prompting—zero-shot and few-shot—to evaluate the performance of both approaches under different conditions. In the zero-shot setting, only an English *task instruction* is provided alongside the input in the target language. The text in Figure 3.9, which outlines the tag set information, serves as the instruction in our experiments. In few-shot prompting, we supplement the prompt with a few English demonstrations, structured according to the prompt template of each method. For *Decom*, we randomly select an example for each tag type from the English training set to create a demonstration. For a fair comparison, the same number of demonstrations is used for the *Iter* baseline. We refer to Appendix B for the details of the prompts used in the experiments.

Evaluation Methods We contrast two evaluation methodologies for prompting in our experiments. The *probability-based* method leverages the model's output logits to retrieve the probability distribution over the tag set, subsequently identifying the label with the highest probability.

[5] https://huggingface.co/datasets/teknium/OpenHermes-2.5

In case the label word is tokenized into subtokens, we use the first subtoken to serve as the label word, following previous work (Zhao et al., 2021; Wang et al., 2023c). The *generation-based* method directly compares the content generated by the LLM with the gold label.

We use the weighted average F1 scores for different tags as our evaluation metric. All experiments were conducted on a server equipped with 4 `A100-SXM4-80GB` GPUs.

3.3.4 Results and Analysis

	Zero-shot		**Few-shot**		**Avg.**
	en	**mult.**	**en**	**mult.**	
`LLaMA2-7B`					
Iter (prob.)	33.1	27.2	68.0	48.6	44.2
Decom (prob.)	**58.2**	**43.2**	**74.7**	**50.5**	**56.7**
Decom (gen.)	53.8	40.4	62.1	45.8	50.5
`LLaMA2-13B`					
Iter (prob.)	47.6	37.4	68.0	52.6	51.4
Decom (prob.)	**67.3**	**54.7**	**77.3**	**54.5**	**63.5**
Decom (gen.)	59.2	48.7	65.3	48.3	55.4
`Mistral-7B`					
Iter (prob.)	**65.2**	54.3	80.2	58.9	64.7
Decom (prob.)	63.6	**61.8**	**85.0**	**64.4**	**68.7**
Decom (gen.)	45.3	48.7	81.4	63.0	59.6

Table 3.23: Overall results of iterative and decomposed prompting methods on POS tagging tasks in zero- and few-shot settings, with F1 score reported. *prob.* denotes probability-based evaluation, while *gen.* signifies generation-based evaluation. **en** indicates the results for English, and **mult.** represents the average F1 score across the other 37 languages. The best performance in each setting is highlighted in **bold**.

We evaluate the performance of iterative and decomposed prompting for English and multilingual POS tag labeling tasks under zero- and few-shot settings. Our goal is (1) to validate the benefits of decomposed prompting in comparison to the baseline method (§3.3.4), and (2) to explore the extent to which decomposed prompting captures multilingual linguistic structure knowledge from the English-centric LLMs (§3.3.5).

3.3.4.1 Main Findings

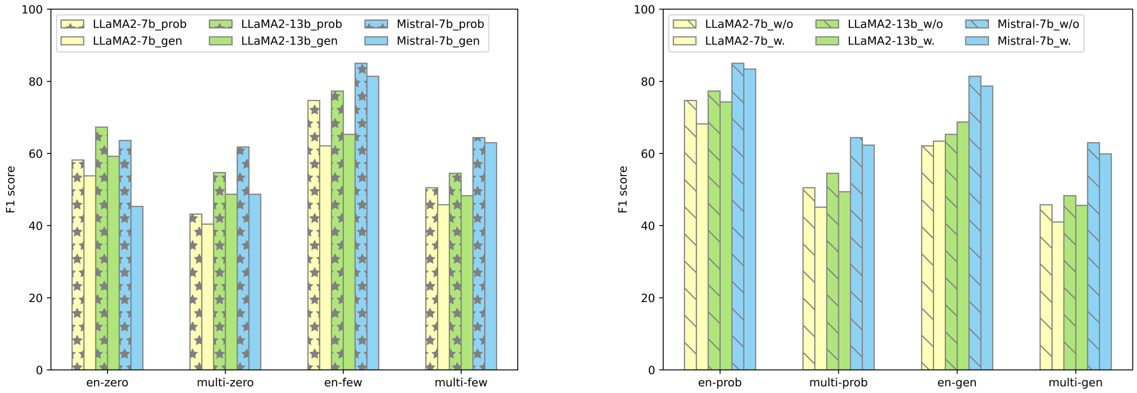
The overall results for English-centric LLMs, as detailed in Table 3.23, demonstrate that our proposed decomposed prompting obviously outperforms the iterative prompting baseline across both zero- and few-shot settings, in both English and multilingual evaluations. This trend holds true for all three English-centric models tested, with the sole exception in the zero-shot setting for the English evaluation with the Mistral-7B model, where *Decom* slightly lags behind *Iter* (63.6 vs. 65.2). In addition to superior performance, decomposed prompting offers enhanced

	BLOOMZ	LLaMA2	Mistral	Avg.
zero-shot	3.2×	2.5×	1.4×	2.4×
few-shot	9.2×	7.9×	3.1×	6.7×

Table 3.24: The ratio by which the inference is accelerated for *Decom* promoting compared to *Iter* prompting.

efficiency during inference, especially with few-shot prompting. As demonstrated in Table 3.24, our proposed method achieves, on average, a 2.4-fold increase in speed compared to the baseline in the zero-shot prompting setting and a 6.7-fold increase in the few-shot setting. The efficiency advantage is less obvious with Mistral, owing to Mistral's implementation of a modified attention mechanism designed to enhance inference efficiency.

3.3.4.2 Ablation Study



(a) Effect of evaluation type

(b) Effect of instruction

Figure 3.11: Results from the ablation study examining the impact of generation-based evaluation method and the inclusion of instruction in prompts across various models and settings. "w/o" denotes the absence of instruction, while "w." signifies the usage of instruction.

We performed an ablation study to investigate two factors: the evaluation method used and the type of instructional prompts. Figure 3.11 presents the outcomes of the ablation study across various model architectures and experimental settings.

Probability-Based vs. Generation-Based We observe in Figure 3.11(a) that the probability-based approach consistently outperforms the generation-based method for both LLaMA2 versions and the Mistral model. This trend is evident in both the English and multilingual tasks, under zero-shot and few-shot conditions. Generally, the performance in few-shot conditions is better than in zero-shot conditions. Notably, in the Mistral-7B model, the gap between the probability-based and generation-based methods narrows in the few-shot condition. The difference between probability- and generation-based evaluation might be that the generation method is able to generate predictions which are invalid POS tags and therefore counted as incorrect whereas this does not occur with the probability-based method.

Effect of Instruction Figure 3.11(b) shows that the inclusion of an instruction in prompts has a variable impact across different models and evaluation methods. In probability-based evaluation, the presence of an instruction leads to a noticeable decrease in F1 scores for all models in both English and multilingual tasks. In generation-based evaluation, we also observe some performance decrease in most cases. This suggests that the LLMs better understand the linguistic structure task from the demonstrations than from a task description in natural language.

3.3.4.3 Case Study

Case 1	
Input:	Die Lage mitten im Niederdorf , wo Abends am meisten los ist , ist wirklich sensationel .
Gloss:	the location middle in Niederdorf , where evenings at most happening is , is really sensational .
True:	DET NOUN ADV ADP PROPN PUNCT ADV ADV ADP PRON ADV VERB PUNCT AUX ADV ADJ PUNCT
***Iter*:**	DET NOUN ADV ADP PROPN PUNCT PRON PROPN ADP ADP NOUN AUX PUNCT VERB ADP ADJ PUNCT
***Decom*:**	DET NOUN ADV ADP PROPN PUNCT ADV PROPN ADP ADP ADV VERB PUNCT PRON ADP ADJ PUNCT
Case 2	
Input:	Damit bestätigten die beiden Konzerne Gerüchte , die seit gestern weltweit die Börsianer in Unruhe versetzten .
Gloss:	thus confirmed the both corporations rumors , which since yesterday worldwide the stock-participants in unrest put .
True:	ADV VERB DET ADJ NOUN NOUN PUNCT PRON ADP ADV ADJ DET NOUN ADP NOUN VERB PUNCT
***Iter*:**	ADP VERB PRON ADJ PROPN NOUN PUNCT PRON ADP ADP ADP PRON PROPN ADP NOUN VERB PUNCT
***Decom*:**	ADV VERB PRON PRON NOUN NOUN PUNCT PRON ADP ADV ADV PRON NOUN ADP NOUN VERB PUNCT

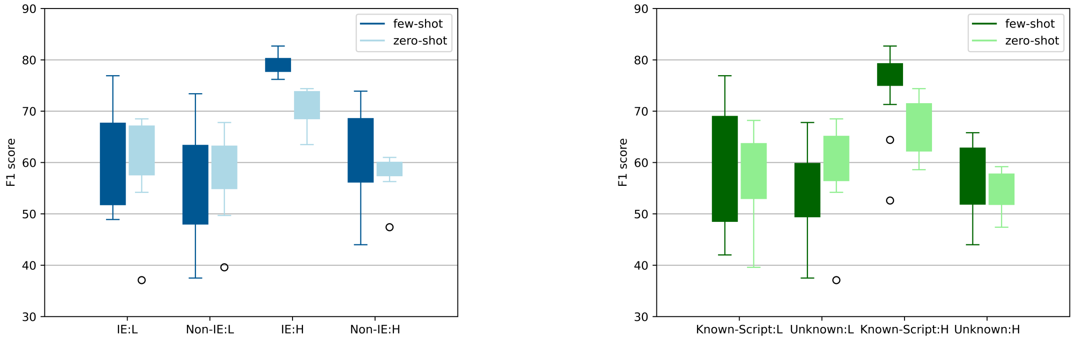
Table 3.25: Comparative analysis of the outputs from *iterative* and *decomposed* prompting methods using selected German examples (***de***) with Mistral. Key tokens and tags are highlighted in red.

The case study presented in Table 3.25 offers a revealing comparative analysis of the iterative and decomposed prompting methods, using selected German examples processed with the Mistral model.

In Case 1, we observe error propagation in the iterative prompting method. The iterative approach incorrectly tags the word “los” as a noun, which subsequently appears to affect the tagging of “ist” as an auxiliary verb rather than the correct tag (`VERB`), as the tag of the copula verb “ist” depends on the constituent linked to it. This illustrates a fundamental weakness of the iterative method: a single misprediction can adversely influence subsequent predictions and lead to a series of errors.

Case 2 reveals a limitation inherent to decomposed prompting, specifically when a word appears multiple times with varying syntactic functions. Take the word “die” in the given case as an example. This word alternates between a definite article and a relative pronoun. The current design of decomposed prompting lacks the sophistication to discern the distinct grammatical roles that a recurring word can play, consequently assigning the tag `PRON` for all instances of "die". Notably, iterative prompting does not resolve this issue in the given example either. This observation underscores the challenge of developing a prompting mechanism capable of context-sensitive discrimination, an area where both decomposed and iterative prompting methods are yet to evolve.

3.3.5 Multilinguality Investigation



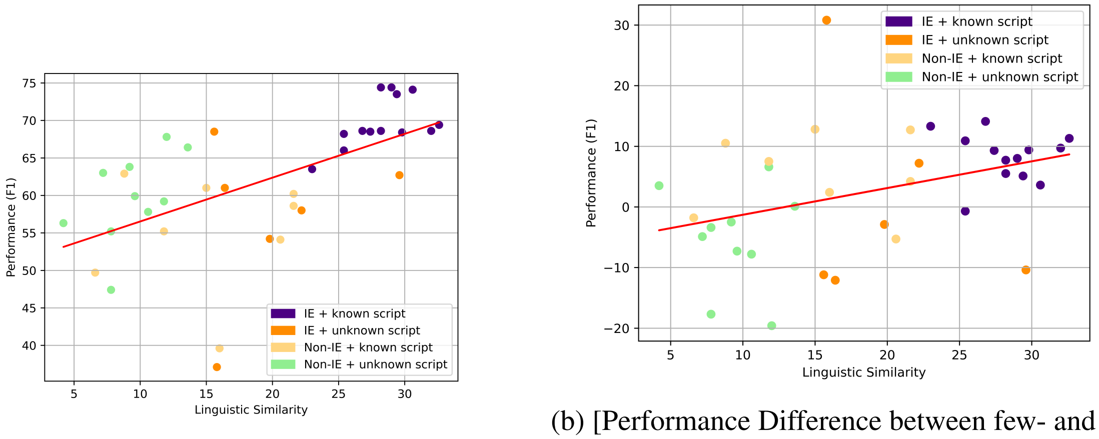
(a) Language Family

(b) Script

Figure 3.12: Analysis of decomposed promoting performance grouped by language family (a) and script type (b) under zero- and few-shot settings on Mistral. "IE" refers to the Indo-European language family. "L" (Low) represents languages that constitute less than 0.005% of the pretraining corpus, while "H" (High) denotes all other languages.

Multilingual Performance Figure 3.12 provides a stratified view of decomposed prompting performance by language family and script, under both zero- and few-shot settings on the Mistral model. The results indicate that Indo-European languages generally achieve higher F1 scores compared to their non-Indo-European counterparts. Notably, the presence of few-shot examples consistently improves the overall performance across all categories, but the box plot also shows that some languages are negatively impacted by the use of English demonstrations. English-centric LLMs are adept at tokenizing words from Latin or Cyrillic scripts into subtokens. For scripts less familiar to these models, they often default to breaking down the text into UTF-8 encodings, which may lead to suboptimal representations for languages using these less common scripts. Thus, to capture a more nuanced understanding of LLM performance across linguistic varieties, we categorize languages not only by family but also by script type. Figure 3.12(b) illustrates that, in both few-shot and zero-shot settings, languages with known scripts tend to yield better performance than unknown scripts. An exception to this trend is observed among the language group with smaller corpora in the zero-shot setting.

To further understand the impact of English demonstrations on languages with varied properties in multilingual prompting, we delve deeper into the cross-lingual transferability of English-centric LLMs and conduct a detailed analysis of individual language performance. We begin



(a) Few-shot

(b) [Performance Difference between few- and zero-shot

Figure 3.13: Panorama of Mistral model's per-language performance. Each node symbolizes a distinct language. (a) shows the few-shot performance, and (b) shows the difference between few- and zero-shot performance for each language.

by quantifying the linguistic proximity of each tested language to English. This was achieved by calculating the cosine similarity between language vectors (Littell et al., 2017) that incorporate syntactic, phylogenetic, and geographic attributes, among others. We follow the method introduced in the previous section (§3.2.5.2) to calculate the language similarity scores. Based on these, we use a rank-based similarity score to average the rank of languages in each feature dimension. Table 3.26 illustrates the computation details. From Figure 3.13, we observe that the performance gain from few-shot prompting is more substantial for languages that are linguistically closer to English, as indicated by the upward trend on the right side of the plot. Remarkably, languages distant from English may even experience a decline in performance when using English demonstrations.

3.3.5.1 English-Centric vs. Multilingual LLMs

Table 3.27[6] shows that English-centric LLMs outperform their multilingual counterparts of comparable size by a considerable margin. This superiority, however, is primarily attributed to their proficiency with the knowledge of English linguistic structures. For languages distant to English and hardly encountered by the English-centric LLMs, such as `el`, `ta`, `te`, `yo`, BLOOMZ and mTk surpass their English-centric counterparts, with mTk exhibiting enhanced performance across as many as 11 linguistically distant languages. Full results are provided in Appendix B. The observations suggest that multilingual LLMs may possess more robust cross-lingual transferability, but are constrained by their inferior base capabilities.

[6] As an LLM of encoder-decoder structure, mTk is not amenable to direct application of our *prob.* evaluation designed for decoder-only models; thus, we resort to *gen.* instead.

	syn.	syn_rank	pho.	pho_rank	inv.	inv_rank	fam.	fam_rank	geo.	geo_rank	rank_score
eng-nld	92.43	37	81.83	18	76.28	36	44.51	35	99.96	37	32.6
eng-deu	90.26	36	80.60	15	78.68	37	54.49	37	99.76	35	32.0
eng-ukr	84.73	32	85.83	32	74.91	33	15.03	30	99.28	26	30.6
eng-por	84.24	31	90.46	35	74.03	28	10.14	22	99.68	33	29.8
eng-ell	78.31	25	95.35	37	74.74	32	15.03	32	98.96	22	29.6
eng-pol	78.64	26	85.83	29	74.09	29	15.03	31	99.63	32	29.4
eng-bul	85.78	35	85.83	30	74.38	30	13.73	27	99.01	23	29.0
eng-ita	85.78	34	85.83	28	72.94	26	11.21	23	99.53	30	28.2
eng-rus	81.18	29	85.83	31	74.63	31	16.80	33	95.81	17	28.2
eng-ron	79.60	27	90.46	34	73.42	27	11.89	24	99.22	25	27.4
eng-spa	82.16	30	85.83	27	72.83	25	9.71	21	99.59	31	26.8
eng-lit	69.33	18	80.42	14	75.58	34	19.39	34	99.44	27	25.4
eng-afr	84.94	33	81.83	17	75.91	35	50.46	36	86.84	6	25.4
eng-fra	81.18	28	75.28	7	72.24	24	9.71	20	99.93	36	23.0
eng-est	77.35	24	85.83	25	70.81	19	0.23	15	99.45	28	22.2
eng-hun	69.40	19	85.83	24	70.66	18	0.33	18	99.46	29	21.6
eng-fin	71.08	21	87.05	33	70.00	17	0.19	13	99.19	24	21.6
eng-eus	62.36	13	85.29	21	70.00	16	3.33	19	99.76	34	20.6
eng-urd	61.63	12	85.83	26	71.98	23	12.71	25	92.54	13	19.8
eng-mar	56.50	8	80.42	13	71.57	22	13.73	28	89.80	11	16.4
eng-wol	63.92	14	85.83	23	69.73	15	0.17	10	96.24	18	16.0
eng-hin	61.63	11	78.35	10	70.91	20	12.71	26	91.10	12	15.8
eng-fas	50.03	3	78.35	11	70.94	21	13.73	29	94.23	14	15.6
eng-ind	72.66	22	90.92	36	67.09	12	0.12	4	79.16	1	15.0
eng-heb	75.15	23	72.55	5	69.10	14	0.13	6	97.16	20	13.6
eng-ara	65.11	16	70.09	3	68.38	13	0.15	9	97.04	19	12.0
eng-tur	50.68	4	81.83	16	67.09	11	0.14	7	98.25	21	11.8
eng-zho	71.08	20	72.55	4	66.94	10	0.33	16	88.42	9	11.8
eng-kaz	44.77	1	83.64	19	66.59	9	0.14	8	95.22	16	10.6
eng-vie	66.04	17	78.35	9	65.81	8	0.19	11	85.25	3	9.6
eng-tel	52.07	6	80.42	12	64.76	4	0.19	14	89.18	10	9.2
eng-tgl	60.89	10	85.83	22	64.76	5	0.13	5	82.15	2	8.8
eng-tam	51.36	5	85.29	20	64.37	3	0.11	3	87.95	8	7.8
eng-kor	55.29	7	74.65	6	63.83	2	0.33	17	86.93	7	7.8
eng-tha	63.95	15	78.35	8	65.40	7	0.11	2	85.25	4	7.2
eng-yor	60.04	9	66.77	2	65.29	6	0.10	1	94.98	15	6.6
eng-jpn	50.03	2	66.77	1	56.88	1	0.19	12	85.65	5	4.2

Table 3.26: Details of language similarity computation.

	Zero-shot		Few-shot		Avg.
	en	mult.	en	mult.	
LLaMA2-7B	58.2	43.2	74.7	50.5	56.7
BLOOMZ-7B	20.6	17.6	44.1	36.2	29.6
LLaMA2-13B	59.2	48.7	65.3	48.3	55.4
mTk-13B	47.6	43.1	57.3	44.7	48.2

Table 3.27: Performance Comparison of English-Centric and Multilingual LLMs. The results of 7B model group are from settings *Decom+prob.*, while the results of 13B model group are from settings *Decom+gen.*

3.3.6 Discussion

Access to LLM Internal Representations Our ablation study empirically proves that the probability-based evaluation method more accurately reflects the multilingual understanding ability of LLMs than the generation-based method, which merely relies on the output text. However, probability-based evaluation relies on the availability of model output logits, a requirement readily met by open-source LLMs, but not by many other LLMs. This availability of the internal representations of open-source LLMs facilitates intriguing research avenues, for instance, the interpretability of LLM behavior (Saha et al., 2023) and the application of LLMs for Bayesian inference (Li et al., 2023a). Hu and Levy (2023) have underscored that direct probability measurement is indispensable in the context of prompting studies. To foster a more transparent and collaborative research environment, better access to the internal workings of LLMs is essential.

LLM's Path to Multilinguality This work analyzed the nuances of multilingual performance across different types of LLMs. The exploration of the multilinguality in English-centric LLMs holds significant practical values, particularly when encountering scenarios that demand a model equipped with robust multilingual skills for tasks like reasoning and commonsense understanding. This investigation is pivotal in guiding the decision-making process regarding the foundational model selection for such tasks. The critical question is whether it is more advantageous to commence with an English-centric LLM, which may offer superior understanding abilities in English, and then endeavor to extend these capabilities to additional languages, or to opt for a multilingual LLM that boasts broader language coverage and enhanced multilingual transferability, albeit potentially at the expense of more refined (English) language understanding abilities. Researchers and practitioners should carefully consider the trade-offs between linguistic breadth and depth of language understanding.

Limitations of Decomposed Prompting As discussed in the analysis part, our proposed *decomposed prompting* strategy struggles if the same word occurs twice in a sentence with different POS tags. Besides, the efficiency of decomposed prompting suffers as the length of the input sequence and the complexity of the task increase. Our study uses decomposed prompting methods for part-of-speech (POS) tagging as a means to evaluate the multilingual structural knowledge of English-centric Large Language Models (LLMs). This provides a foundational assessment of the models' capabilities. Nevertheless, the scope for extending this methodology to probe more intricate aspects of linguistic structure is substantial. Future research could beneficially apply decomposed prompting to the analysis of complex linguistic phenomena, including sentence chunking and syntactic parsing, to gain a deeper understanding of the nuanced capabilities of LLMs in processing and understanding language.

3.3.7 Sum-Up

In conclusion, our investigation into the multilingual capabilities of English-centric LLMs through the lens of decomposed prompting has yielded significant findings. By systematically dissecting the sequence labeling process into discrete, token-level prompts, we have demonstrated that these models possess a considerable understanding of linguistic structure that extends beyond their predominant English training. Our method outperforms existing iterative prompting techniques in both zero- and few-shot settings, highlighting the efficiency and accuracy of decomposed prompting. The empirical evidence suggests that while English-centric LLMs can effectively engage in multilingual tasks, their performance is nuanced and influenced by the linguistic proximity to English and the design of the prompting strategy. This work not only advances the field of NLP by enhancing our understanding of LLMs' cross-lingual transfer capabilities but also opens avenues for future research to further improve the inclusivity and adaptability of language models for a diverse range of languages and tasks.

3.4 In-Context Learning for Cross-Lingual Knowledge Editing

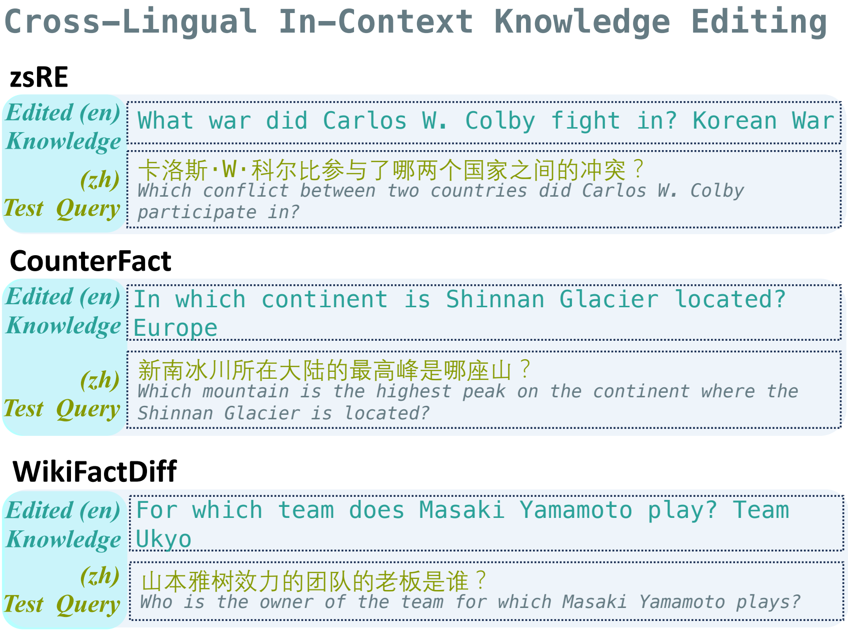
This section corresponds to the following work:

> **Ercong Nie*** Bo Shao*, Mingyang Wang, Zifeng Ding, Helmut Schmid, Hinrich Schütze. 2025. BMIKE-53: Investigating Cross-Lingual Knowledge Editing with In-Context Learning. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025. Volume 1: Long Papers)
> * equal contributions.

Declaration of Co-Authorship. I conceived the idea of building a comprehensive multilingual knowledge editing benchmark and proposed the research question of investigating the performance of cross-lingual knowledge editing with in-context learning paradigms. I did the data collection, data preprocessing, and benchmark construction. Bo Shao set up the pipeline to evaluate in-context knowledge editing on the proposed benchmark. He also adapted the code to run gradient-based baseline methods. I ran part of the experiments and conducted the data analysis work. Besides, I wrote the manuscript of this paper. Mingyang Wang and Zifeng Ding contributed by attending the discussions and providing feedback. Both also contributed to the formulation of this research idea and the research question. Helmut Schmid and Hinrich Schütze supervised the course of the project.

Summary of This Section

This section shifts the focus to prompt-based learning for multilingual factual knowledge. We investigate the application of in-context learning for cross-lingual knowledge editing. For this purpose, we introduce BMIKE-53, a comprehensive benchmark for cross-lingual in-context knowledge editing (IKE) across 53 languages, unifying three knowledge editing (KE) datasets: zsRE, CounterFact, and WikiFactDiff. Cross-lingual KE, which requires knowledge edited in one language to generalize across others while preserving unrelated knowledge, remains underexplored. To address this gap, we systematically evaluate IKE under zero-shot, one-shot, and few-shot setups, incorporating tailored metric-specific demonstrations. Our findings reveal that model scale and demonstration alignment critically govern cross-lingual IKE efficacy, with larger models and tailored demonstrations significantly improving performance. Linguistic properties, particularly script type, strongly influence performance variation across languages, with non-Latin languages underperforming due to issues like language confusion.



What war did Carlos W. Colby fight in? Korean War

卡洛斯·W·科尔比参与了哪两个国家之间的冲突？
Which conflict between two countries did Carlos W. Colby participate in?

In which continent is Shinnan Glacier located? Europe

新南冰川所在大陆的最高峰是哪座山？
Which mountain is the highest peak on the continent where the Shinnan Glacier is located?

For which team does Masaki Yamamoto play? Team Ukyo

山本雅树效力的团队的老板是谁？
Who is the owner of the team for which Masaki Yamamoto plays?

Figure 3.14: Examples of cross-lingual in-context knowledge editing.

3.4.1 Background and Motivation

Large language models (LLMs) have demonstrated remarkable abilities to encode vast amounts of knowledge during pre-training, enabling them to perform well across a range of tasks (Min et al., 2022; Zhang et al., 2023a; Zhou et al., 2024). However, this knowledge remains static, becoming outdated as the world evolves, necessitating mechanisms to update models with new facts while preserving their overall performance (Cao et al., 2021; Dhingra et al., 2022). Traditional approaches, such as fine-tuning, are computationally expensive and impractical for closed-

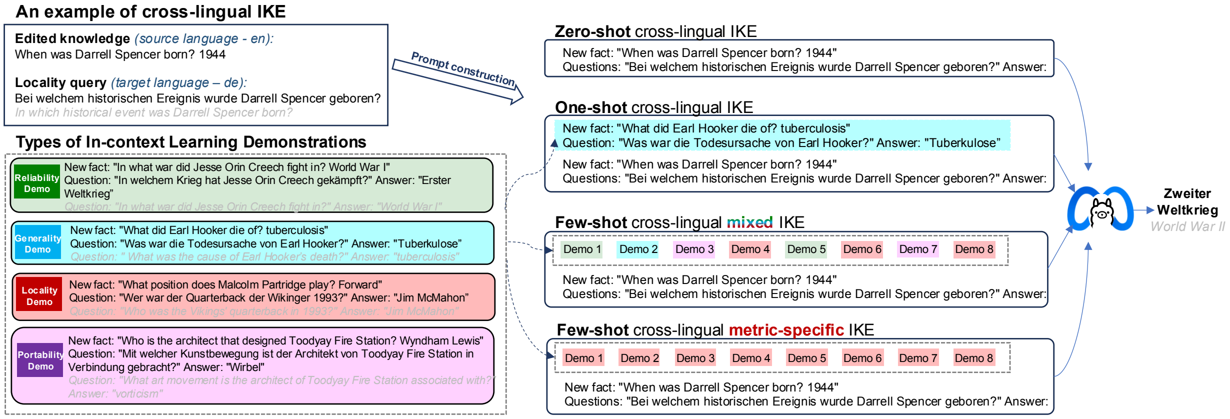

An example of cross-lingual IKE
Edited knowledge (source language - en):
When was Darrell Spencer born? 1944
Locality query (target language – de):
Bei welchem historischen Ereignis wurde Darrell Spencer geboren?
In which historical event was Darrell Spencer born?
Prompt construction
Types of In-context Learning Demonstrations
Reliability Demo
New fact: "In what war did Jesse Orin Creech fight in? World War I"
Question: "In welchem Krieg hat Jesse Orin Creech gekämpft?" Answer: "Erster Weltkrieg"
Question: "In what war did Jesse Orin Creech fight in?" Answer: "World War I"
Generality Demo
New fact: "What did Earl Hooker die of? tuberculosis"
Question: "Was war die Todesursache von Earl Hooker?" Answer: "Tuberkulose"
Question: " What was the cause of Earl Hooker's death?" Answer: "tuberculosis"
Locality Demo
New fact: "What position does Malcolm Partridge play? Forward"
Question: "Wer war der Quarterback der Wikinger 1993?" Answer: "Jim McMahon"
Question: "Who was the Vikings' quarterback in 1993?" Answer: "Jim McMahon"
Portability Demo
New fact: "Who is the architect that designed Toodyay Fire Station? Wyndham Lewis"
Question: "Mit welcher Kunstbewegung ist der Architekt von Toodyay Fire Station in Verbindung gebracht?" Answer: "Wirbel"
Question: "What art movement is the architect of Toodyay Fire Station associated with?" Answer: "vorticism"
Zero-shot cross-lingual IKE
New fact: "When was Darrell Spencer born? 1944"
Questions: "Bei welchem historischen Ereignis wurde Darrell Spencer geboren?" Answer:
One-shot cross-lingual IKE
New fact: "What did Earl Hooker die of? tuberculosis"
Question: "Was war die Todesursache von Earl Hooker?" Answer: "Tuberkulose"
New fact: "When was Darrell Spencer born? 1944"
Questions: "Bei welchem historischen Ereignis wurde Darrell Spencer geboren?" Answer:
Few-shot cross-lingual mixed IKE
Demo 1 Demo 2 Demo 3 Demo 4 Demo 5 Demo 6 Demo 7 Demo 8
New fact: "When was Darrell Spencer born? 1944"
Questions: "Bei welchem historischen Ereignis wurde Darrell Spencer geboren?" Answer:
Few-shot cross-lingual metric-specific IKE
Demo 1 Demo 2 Demo 3 Demo 4 Demo 5 Demo 6 Demo 7 Demo 8
New fact: "When was Darrell Spencer born? 1944"
Questions: "Bei welchem historischen Ereignis wurde Darrell Spencer geboren?" Answer:
Zweiter Weltkrieg
World War II

Figure 3.15: Cross-Lingual IKE setups and demonstration types.

source or large-scale models (Dai et al., 2022). These limitations have motivated the emergence of knowledge editing (KE)—a technique for selectively modifying LLMs to incorporate new knowledge while maintaining the integrity of unrelated knowledge (Zhang et al., 2024b).

Recent advancements in KE have explored gradient-free methods inspired by in-context learning (ICL), where LLMs learn through prompts and demonstrations without requiring parameter updates (Zheng et al., 2023). These methods are efficient and particularly suitable for scenarios where direct access to model parameters is restricted. However, existing gradient-free KE research primarily focuses on monolingual settings, leaving the potential for cross-lingual KE largely unexplored. Cross-lingual KE, a more challenging task, as illustrated in Figure 3.14, requires knowledge edited in one language (e.g., English) to generalize effectively to semantically equivalent queries across diverse target languages while preserving unrelated knowledge.

This study addresses the critical gap in cross-lingual knowledge editing by proposing a comprehensive and multidimensional investigation of in-context knowledge editing (IKE) methods. We introduce **BMIKE-53**, a multilingual benchmark spanning 53 languages and integrating three representative KE datasets: zsRE, which evaluates regular fact modifications; CounterFact, which examines counterfactual knowledge updates; and WikiFactDiff (WFD), which assesses real-world, temporally dynamic knowledge updates. This benchmark is the most comprehensive multilingual KE resource to date, unifying diverse KE datasets into a consistent format and expanding them into multiple languages using LLM-assisted translation. The wide linguistic coverage allows us to systematically analyze cross-lingual differences and their underlying causes.

To evaluate cross-lingual IKE, we implement zero-shot, one-shot, and few-shot setups to explore the impact of demonstration quality and quantity on performance (see Figure 3.15). Notably, we propose two few-shot setups: 8-shot mixed demonstrations, which expose the model to diverse query types, and 8-shot metric-specific demonstrations, which target specific query types like locality or portability to enhance performance. These setups allow us to analyze the interplay between demonstration strategies, query types, and cross-lingual transfer. Our findings show that larger models and tailored demonstrations significantly improve performance, especially for complex queries. Linguistic properties, such as syntactic and phonological similarity with English, positively influence performance, while language family has no significant impact. Instead, script type emerges as a critical factor, with non-Latin languages underperforming due to issues like language confusion, where models generate answers in English instead of the target language.

In summary, our contributions are as follows: **i)** We introduce BMIKE-53, the most comprehensive multilingual KE benchmark, covering 53 languages and three diverse KE datasets, which serves as a foundation for evaluating cross-lingual KE methods. **ii)** We extensively evaluate gradient-free cross-lingual KE methods under various IKE setups, providing valuable insights into the effectiveness of in-context learning for cross-lingual knowledge editing. **iii)** We conduct a detailed analysis of factors influencing cross-lingual KE performance, uncovering the impact of linguistic properties, script types, and language confusion on cross-lingual knowledge transfer.

3.4.2 Existing Knowledge Editing Methods

Traditional KE methods are primarily gradient-based. They typically introduce additional trainable parameters, such as MEND (Mitchell et al., 2021) and SERAC (Mitchell et al., 2022)—or edit specific parameters of the original model, as in ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023). However, these methods have high computational demands and are difficult to scale. Recent studies have explored gradient-free KE methods for LLMs, inspired by the in-context learning (ICL) paradigm, where LLMs learn from prompts and demonstrations without parameter updating, such as IKE (Zheng et al., 2023), MeLLo (Zhong et al., 2023), and ICE (Cohen et al., 2024). Given the multilingual in-context learning capabilities of English-centric LLMs (Lai et al., 2023a; Nie et al., 2024; Zhang et al., 2024a), the potential for cross-lingual KE appears promising. However, current gradient-free KE methods have primarily been explored within monolingual contexts.

Recent cross-lingual KE work largely employs gradient-based methods (Xu et al., 2023; Wang et al., 2023a; Beniwal et al., 2024; Wei et al., 2024). A notable gradient-free work is ReMaKE (Wang et al., 2024a), a cross-lingual retrieval-augmented KE method. However, their method is specifically applied to a rather special KE scenario—batch edit. In this setting, multiple knowledge pieces, such as the entire knowledge base, are edited simultaneously. Our work diverges from these existing approaches in the following key aspects. Regarding task setup, ReMaKE employs a cross-lingual retrieval-augmented strategy tailored for batch edits, allowing simultaneous modifications of multiple knowledge pieces, such as an entire knowledge base. Conversely, our approach focuses on individual knowledge edits, which do not involve cross-lingual retrieval. This fundamental difference sets our approach apart and addresses a unique aspect of knowledge editing. In prompt engineering, when designing the demonstrations of ICL, ReMaKE uses translation pairs of source and target language facts to make up the cross-lingual demonstrations, from which the model cannot learn real cross-lingual knowledge editing competencies like portability and locality. In contrast, our MIKE method provides four different types of cross-lingual ICL demonstrations. From these demonstrations, the model can effectively learn real cross-lingual knowledge editing.

3.4.3 BMIKE-53

BMIKE-53 spans a wide range of knowledge editing perspectives, from artificial to realistic scenarios, and provides a solid foundation for evaluating cross-lingual KE methods. Additionally, with coverage of 53 languages, it stands as the most comprehensive multilingual KE benchmark to date.

Task	#Test	Q-Len.	A-Len.	#Lang.
zsRE	743	9.02	2.02	53
CounterFact	1,031	5.97	1.00	
WFD	784	4.71	2.55	

Table 3.28: Statistics of BMIKE-53. Q/A-Len.: Average Text Length of Query/Answer.

3.4.3.1 Datasets

As shown in Figure 3.14, BMIKE-53 is constructed from three monolingual KE datasets: **zsRE**, **CounterFact**, and **WikiFactDiff (WFD)**. Each dataset was selected to represent a distinct perspective of knowledge editing, ensuring the benchmark comprehensively evaluates diverse KE scenarios.

The **zsRE** dataset, originally introduced by Levy et al. (2017), was designed for zero-shot relation extraction and later adapted by De Cao et al. (2021) and Mitchell et al. (2021) for knowledge editing tasks. zsRE focuses on regular, well-defined knowledge items, making it an ideal baseline for evaluating the reliability and generality of KE methods.

The **CounterFact** dataset, introduced by Meng et al. (2022), is designed to evaluate the ability of models to update knowledge with counterfactual (false) facts. Each entry in CounterFact represents a knowledge triple that has been altered to reflect a hypothetical or fabricated scenario. CounterFact is particularly valuable for assessing the locality of KE methods, as it requires precise updates to counterfactual knowledge without unintended side effects.

The **WikiFactDiff (WFD)** dataset, introduced by Khodja et al. (2024), focuses on real-world, temporally recent knowledge updates. Derived from WikiData (Vrandečić and Krötzsch, 2014), WFD captures changes to knowledge triples that reflect actual updates in the real world, such as changes in political leadership, scientific discoveries, or other evolving facts. WikiFactDiff is essential for assessing the real-world applicability of KE methods, as it introduces the challenge of updating models with temporally recent and realistic knowledge changes.

3.4.3.2 Benchmark Construction

The construction of the BMIKE-53 benchmark involves three key steps: unifying data formats, multilingual expansion, and quality control. These steps ensure that the benchmark is consistent, multilingual, and of high quality, enabling robust evaluation of cross-lingual knowledge editing (KE) methods.

> **system:**
> You are an intelligent multilingual translation assistant that can structurally translate English text in a fixed data format into 52 different languages.
>
> **user:**
> Translate the following JSON data item from English to {`target language`}. Keep the original JSON format and structure, translating only the text in the values while keeping the key names unchanged. Output only in plain text without additional formatting text. Use double quotes for key and value names and add the escape character for quote marks in the text of value: {`json_data_item`}

Figure 3.16: Prompt template for `GPT-4o` as translation assistant.

Unifying Data Formats To ensure consistency across the three datasets, we standardized their formats into a unified structure to create a cohesive English base dataset. Each data item in the unified format includes an edited knowledge item and four types of test queries: reliability, generality, locality, and portability. The portability queries for zsRE and CounterFact were adopted from the work of Yao et al. (2023), while for WFD, we extracted a knowledge graph from the original WFD dataset and performed one-hop knowledge reasoning within the graph to generate portability queries. All data items are stored in a JSON format, with a consistent data structure across the three datasets. This unified format facilitates seamless integration into the benchmark and enables efficient LLM-assisted translation in the multilingual expansion process.

Structured Multilingual Expansion with LLMs To create a multilingual benchmark for cross-lingual KE tasks, we expanded the English base data into 52 target languages using LLM-assisted structural translation. The language coverage is adopted from similar multilingual datasets like MLAMA (Kassner et al., 2021) and BMLAMA (Qi et al., 2023). BMIKE-53 encompasses a total of 53 languages. A comprehensive list of these languages can be found in Table 3.29. Additionally, the table outlines the linguistic feature similarities between each target language and English. The similarity scores are calculated as introduced in the previous section (§3.2.5.2).

We employed the GPT-4o model[7] via the OpenAI API for the multilingual expansion. The translation process was guided by a structured prompt template, as displayed in Figure 3.16, which ensured that the JSON format and structure of the data items were preserved during translation. Specifically: Only the text values within the JSON structure were translated, while the key names remained unchanged. The prompt explicitly instructed the model to output the translated data in plain text, adhering to the original JSON format. We compared several machine translation tools, including NLLB-200 and Google Translate API, but found that LLM-assisted translation offered superior performance in terms of:

- Accuracy: LLMs demonstrated better handling of complex linguistic structures and domain-

[7] `gpt-4o-2024-08-06`

lid	language	Family	syn_sim	pho_sim	inv_sim	gen_sim	geo_sim
af	Afrikaans	Indo-European (Germanic)	84.94	81.83	69.10	50.46	86.84
ar	Arabic	Semitic	65.11	70.09	70.81	0.15	97.04
az	Azerbaijani	Turkic	52.00	81.83	67.86	0.19	96.96
be	Belarusian	Indo-European (Slavic)	78.64	85.83	70.42	16.80	99.35
bg	Bulgarian	Indo-European (Slavic)	85.78	85.83	68.38	13.73	99.01
bn	Bengali	Indo-European (Indo-Aryan)	58.36	76.30	74.38	12.71	88.96
ca	Catalan	Indo-European (Romance)	87.30	85.83	75.22	10.64	99.66
ceb	Cebuano	Austronesian	62.17	76.30	75.22	0.13	81.50
cs	Czech	Indo-European (Slavic)	73.99	85.83	66.51	13.73	99.71
cy	Welsh	Indo-European (Celtic)	71.90	81.83	77.85	13.73	99.99
da	Danish	Indo-European (Germanic)	88.01	81.83	77.54	40.90	99.89
de	German	Indo-European (Germanic)	90.26	80.60	76.28	54.49	99.76
el	Greek	Indo-European (Hellenic)	78.31	95.35	64.76	15.03	98.96
es	Spanish	Indo-European (Romance)	82.16	85.83	63.83	9.71	99.59
et	Estonian	Uralic	77.35	85.83	66.94	0.23	99.45
eu	Basque	Isolate	62.36	85.29	56.88	3.33	99.76
fa	Persian	Indo-European (Iranian)	50.03	78.35	72.83	13.73	94.23
fi	Finnish	Uralic	71.08	87.05	70.00	0.19	99.19
fr	French	Indo-European (Romance)	81.18	75.28	74.09	9.71	99.93
ga	Irish	Indo-European (Celtic)	72.01	85.83	69.35	12.71	99.96
gl	Galician	Indo-European (Romance)	80.23	90.46	70.75	10.14	99.65
he	Hebrew	Semitic	75.15	72.55	64.37	0.13	97.16
hi	Hindi	Indo-European (Indo-Aryan)	61.63	78.35	70.91	12.71	91.10
hr	Croatian	Indo-European (Slavic)	83.18	85.83	69.67	12.71	99.50
hu	Hungarian	Uralic	69.40	85.83	74.03	0.33	99.46
hy	Armenian	Indo-European (Satem)	63.03	69.66	68.73	19.39	97.23
id	Indonesian	Austronesian	72.66	90.92	75.58	0.12	79.16
it	Italian	Indo-European (Romance)	85.78	85.83	70.00	11.21	99.53
ja	Japanese	Isolate	50.03	66.77	65.40	0.19	85.65
ka	Georgian	Caucasian	68.50	66.93	62.93	0.19	97.09
ko	Korean	Isolate	55.29	74.65	70.94	0.33	86.93
la	Latin	Indo-European (Romance)	78.27	85.83	76.76	15.03	99.47
lt	Lithuanian	Indo-European (Baltic)	69.33	80.42	74.63	19.39	99.44
lv	Latvian	Indo-European (Baltic)	75.39	81.83	75.22	19.39	99.42
ms	Malay	Austronesian	70.49	90.92	72.49	0.15	80.49
nl	Dutch	Indo-European (Germanic)	92.43	81.83	72.24	44.51	99.96
pl	Polish	Indo-European (Slavic)	78.64	85.83	65.29	15.03	99.63
pt	Portuguese	Indo-European (Romance)	84.24	90.46	78.68	10.14	99.68
ro	Romanian	Indo-European (Romance)	79.60	90.46	73.42	11.89	99.22
ru	Russian	Indo-European (Slavic)	81.18	85.83	64.76	16.80	95.81
sk	Slovak	Indo-European (Slavic)	82.16	85.83	70.66	15.03	99.55
sl	Slovenian	Indo-European (Slavic)	80.59	85.83	75.58	15.03	99.62
sq	Albanian	Indo-European (Other)	79.60	87.05	72.49	33.48	99.19
sr	Serbian	Indo-European (Slavic)	79.60	85.83	72.94	12.71	99.23
sv	Swedish	Indo-European (Germanic)	93.34	81.83	67.98	40.90	99.62
ta	Tamil	Dravidian	51.36	85.29	65.81	0.11	87.95
th	Thai	Kra-Dai	63.95	78.35	74.91	0.11	85.25
tr	Turkish	Turkic	50.68	81.83	66.59	0.14	98.25
uk	Ukrainian	Indo-European (Slavic)	84.73	85.83	74.38	15.03	99.28
ur	Urdu	Indo-European (Indo-Aryan)	61.63	85.83	71.57	12.71	92.54
vi	Vietnamese	Austroasiatic	66.04	78.35	74.74	0.19	85.25
zh-cn	Chinese	Sino-Tibetan	71.08	72.55	69.73	0.33	88.42

Table 3.29: Detailed information of the languages covered by BMIKE-53. The right five columns show the linguistic feature similarities between the target language and English. syn: syntax, pho: phonology, inv: phonetics, gen: phylogenetic, geo: geographic, sim: similarity.

specific terminology.

- Flexibility: LLMs provided greater adaptability for processing structured data formats like

JSON.

- Consistency: The structured translation process ensured that the multilingual data remained aligned with the original English data.

By leveraging LLM-assisted translation, we ensured that the multilingual benchmark maintained high linguistic quality and structural integrity.

Quality Control To ensure the quality of the multilingual expansion, we implemented a rigorous quality control process involving qualitative evaluation and quantitative analysis. We conducted a manual review of sampled sentences by native speakers of selected languages, then we used back-translation techniques to provide an overall assessment of translation quality. Specifically, each translated sentence was back-translated into English, and the BLEU score and semantic similarity between the original and the back-translated English text were calculated. BLEU Score measures the formal similarity between the original and back-translated sentences. Semantic Similarity evaluates the semantic alignment between the original and back-translated sentences using cosine similarity in a sentence embedding space. The results of the back-translation evaluation are shown in Table 3.30.

	zsRE		**CounterFact**		**WFD**	
Lang.	**BLEU**	**Sim.**	**BLEU**	**Sim.**	**BLEU**	**Sim.**
es	0.81	0.94	0.82	0.90	0.81	0.90
vi	0.82	0.93	0.77	0.91	0.78	0.90
ru	0.78	0.91	0.71	0.87	0.72	0.87
zh	0.78	0.89	0.76	0.85	0.76	0.85
de	0.84	0.93	0.82	0.92	0.82	0.92

Table 3.30: Results of Translation Quality Control via Back-Translation.

3.4.4 Experiments

The primary goal of this work is to extensively investigate the performance of cross-lingual in-context knowledge editing (IKE). Using the proposed benchmark BMIKE-53, we aim to explore the factors influencing cross-lingual IKE performance, identify performance tendencies, and analyze variations in cross-lingual behavior across different languages and query types. To achieve this, we first formally define the cross-lingual IKE task and its evaluation framework. We then introduce the different IKE setups and strategies explored in our experiments

3.4.4.1 Task: Cross-Lingual In-Context Knowledge Editing

The Cross-Lingual In-Context Knowledge Editing (IKE) task evaluates a language model's ability to incorporate new knowledge (a fact) in one language and apply it across multiple languages while preserving unrelated knowledge. This task leverages in-context learning (ICL) to guide the model in editing and applying knowledge through demonstrations. Below, we formally define the task and the four types of cross-lingual queries used to evaluate the model's performance.

Task Definition Given a language model $\mathcal{M}$; a new fact represented as a query-answer pair $f = (x_s^*, y_s^*)$ in the source language s, where x_s^* is the query and y_s^* is the corresponding answer; a set $\mathcal{X}_s^*$, which contains x_s^* and other semantically equivalent queries in the source language; the translations of $\mathcal{X}_s^*$ into a target language t, denoted as $\mathcal{X}_t^* = \{I_t(x_s) : x_s \in \mathcal{X}_s^*\}$ where $I^t(\cdot)$ is a translator mapping source language queries to their target language counterparts. The task involves evaluating the model's response to a target language query x_t after incorporating the new fact f. Specifically, the model assigns a probability $P_{\mathcal{M}}(y|x_t, f)$ to an answer y given the query x_t and the fact f. The predicted answer is defined as $Pred(x_t, f) = \text{argmax}_y \mathcal{P}_{\mathcal{M}}(y|x_t, f)$. The goal is for the model to predict the correct translation of the fact answer, $I^t(y_s^*)$, when the query x_t is semantically equivalent to the fact query x_s^*, and to preserve the original knowledge and predict the correct answer y_t for unrelated queries, ensuring no unintended inference from the knowledge editing process.

Cross-Lingual Query Types To comprehensively evaluate the model's cross-lingual knowledge editing capabilities, we define four types of target language queries, each testing a specific aspect of the task. The model should reliably apply the new fact to the exact translation of the original query. A ***reliability*** query x_t is defined as $x_t = I_t(x_s^*)$. ***Generality*** queries test whether the model can generalize the new fact to other semantically equivalent queries in the target language that differ in phrasing or structure. A generality query x_t is defined as $x_t \in \mathcal{X}_t^* \backslash \{I_t(x_s^*)\}$. The expected answer for the reliability and generality query is $Pred(x_t, f) = I_t(y_s^*)$. ***Portability*** queries evaluate whether the model can apply the new fact to related but contextually different queries in the target language. These queries are derived through one-hop knowledge reasoning from the original fact. Let $\mathcal{X}_{s,\text{1-hop}}^*$ denote the one-hop query set in the source language, which includes x_s^* and queries influenced by x_s^* through one-hop reasoning in a knowledge graph. The corresponding target language set is $\mathcal{X}_{t,\text{1-hop}}^* = \{I_t(x_s^*) : x_s \in \mathcal{X}_{s,\text{1-hop}}^*\}$. A portability query x_t is defined as $x_t \in \mathcal{X}_{t,\text{1-hop}}^*$. ***Locality*** queries test whether the model can preserve unrelated knowledge while incorporating the new fact. A locality query x_t is defined as $x_t \notin \mathcal{X}_{t,\text{1-hop}}^*$. The expected answer is $Pred(x_t, f) = y_t$.

3.4.4.2 Cross-Lingual IKE Setup

IKE Demonstrations In the context of in-context learning (ICL), demonstrations are examples provided in the input prompt to guide the model's behavior. For the IKE task, the demonstrations are designed to teach the model how to perform cross-lingual knowledge editing. Formally, the set of demonstrations is defined as $C = \{c_1, \cdots, c_k\}$, where each demonstration c_i consists of a new fact $f' = (x_s', y_s')$ in the source language, a query x_t' in the target language, and the correct answer y_t' in the target language. As an ICL-based KE method, IKE uses demonstrations to guide the model in learning the relationships between the source language fact and the target language queries. As illustrated in Figure 3.15, the demonstrations are designed to reflect the four query types, enabling the model to learn the appropriate behavior for each type. By including examples of cross-lingual queries and their correct answers, the demonstrations help the model generalize the knowledge editing process across languages.

IKE Setup As illustrated in Figure 3.15, we evaluate cross-lingual IKE under four distinct setups: zero-shot, one-shot, few-shot mixed, and few-shot metric-specific. In ***zero-shot*** cross-lingual IKE, the model performs cross-lingual knowledge editing without any demonstrations, relying solely on its pre-trained capabilities. In a ***one-shot*** setup, a single randomly selected demonstration is provided to familiarize the model with the task format. This setup is designed to give the model an overview of the task without significantly aiding its ability to complete the task. ***Few-shot mixed*** cross-lingual IKE includes eight demonstrations of mixed types. The goal is to teach the model cross-lingual knowledge editing by exposing it to diverse query types. To enhance performance on specific query types, the ***few-shot metric-specific*** variation provides eight demonstrations of the same query type as the test target.

3.4.4.3 Experimental Setting

We conduct experiments using two multilingual LLMs: Llama3.2-3B and Llama3.1-8B. These models were selected for their multilingual capabilities and represent different model sizes, allowing us to analyze the impact of model scale on cross-lingual IKE performance. To measure the cross-lingual IKE performance, we compare the predicted answers with the ground-truth answers using the **F1** score and Exact Match (**EM**) metrics, consistent with prior work (Wang et al., 2023a, 2024a). Table 3.31 displays the experiment implementation details. We downloaded the models from HuggingFace[8].

Parameter	Value
Model	`meta-llama/Llama-3.1-8B`, `meta-llama/Llama-3.2-3B`
Max. length	4096
Num. of demonstration	8
Type of demonstration	Reliability, Generality, Locality, Portability
Proportion of demo.	1:3:2:2
GPU Type	NVIDIA A100-SXM4-80GB
Number of GPU	4
Running hours	72

Table 3.31: Experimental Implementation Details.

3.4.5 Multidimensional Analysis of Cross-Lingual IKE

This section provides a multidimensional analysis of cross-lingual IKE performance, focusing on the effects of model size, dataset-specific performance, query type variations, and IKE setup strategies. Using the BMIKE-53 benchmark, we aim to uncover key insights into how these factors influence IKE performance and cross-lingual variations. Table 3.32 shows the overall experimental results.

[8] https://huggingface.co/meta-llama/Llama-3.2-3B and https://huggingface.co/meta-llama/Llama-3.1-8B

Model	Setup	zsRE				CounterFact				WikiFactDiff			
		rel	gen	loc	port	rel	gen	loc	port	rel	gen	loc	port
`Llama3.2-3B`	*zero-shot*	50.16	49.03	5.64	5.41	43.51	40.84	7.53	5.61	58.28	57.41	6.77	3.18
	one-shot	71.57	71.25	7.78	14.97	68.34	68.02	6.58	16.30	66.32	65.93	6.31	3.06
	8-shot (mix)	70.54	70.49	8.89	16.43	70.80	70.33	5.11	13.75	64.97	64.60	6.24	4.00
	8-shot (metric)	70.94	70.91	12.23	22.97	67.57	67.14	31.61	31.20	67.77	67.48	9.14	10.72
`Llama3.1-8B`	*zero-shot*	65.53	64.09	9.76	10.05	63.01	60.59	18.68	11.16	67.84	66.40	10.04	4.15
	one-shot	75.27	74.90	13.36	20.81	71.92	71.29	12.66	21.93	70.53	69.84	7.80	4.15
	8-shot (mix)	74.29	74.00	15.46	25.18	75.15	74.42	11.40	23.74	68.57	67.86	8.27	8.87
	8-shot (metric)	74.86	74.79	16.15	32.86	73.88	73.19	47.55	41.17	71.98	71.34	13.84	14.58

Table 3.32: Main Results. Average cross-lingual IKE performance across 52 languages (F1-score).

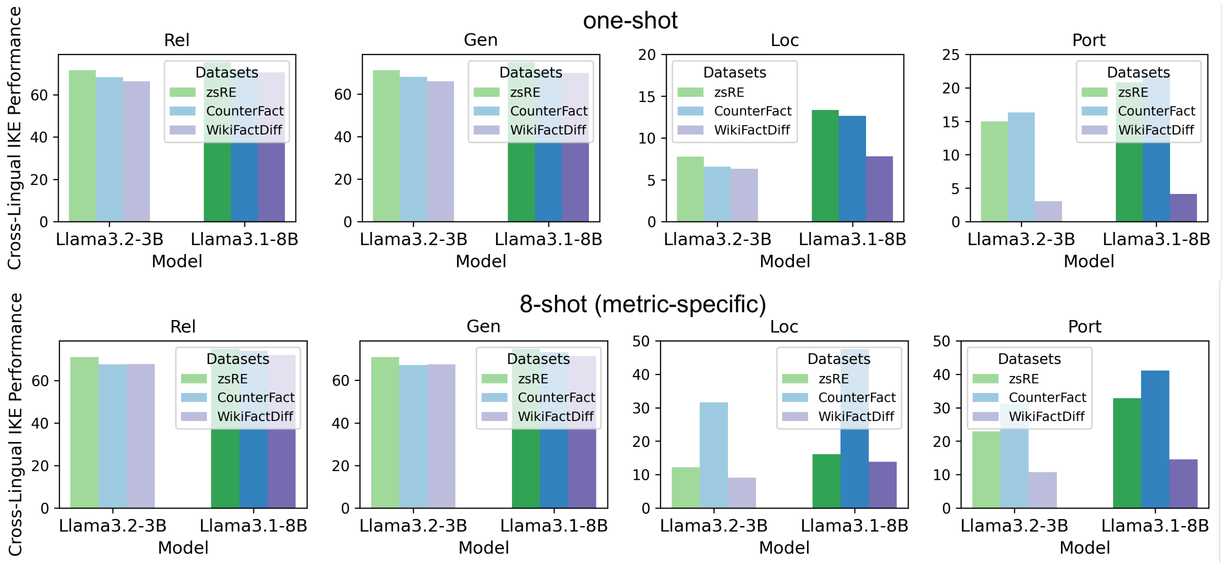


Figure 3.17: Average cross-lingual IKE performance across languages (F1-score).

Effects of Model Scale As shown in Figure 3.17, larger models demonstrate superior cross-lingual IKE capabilities across all experimental configurations. The 8B-parameter model achieves 65.53 F1 score on zsre reliability (rel) in the zero-shot setting, compared to 50.16 for the base 3B model—representing a 15.37-point improvement. This performance gap persists across demonstration strategies, with the 8-shot metric-specific setup yielding 32.86 vs 22.97 portability (port) scores on zsRE. Notably, larger models show particular advantages in handling queries requiring complex multilingual reasoning and knowledge preservation, evidenced by the performance gap of locality (loc) and portability (port) queries.

Dataset-Specific Performance Patterns We observe substantial cross-dataset variance in editing efficacy, especially with loc and port queries, indicative of inherent dataset complexity differences. While WikiFactDiff (WFD) achieves comparable reliability (rel) and generality (gen) scores to zsRE and CounterFact, it shows the lowest port performance. This discrepancy could be attributed to the real-world nature of WFD, where all knowledge–both the original and updated facts–is temporally recent. Portability queries in WFD require the model to reason over a second-order knowledge chain, where the correct answer depends on understanding the relationship between the updated fact and its broader context. CounterFact, on the other hand, benefits the most from metric-specific demonstrations, particularly for locality and portability. The compositional nature of CounterFact's fabricated facts allows the model to leverage targeted demonstrations effectively.

Query-Type Sensitivity The four query types exhibit divergent response patterns to IKE strategies. Figure 3.18 shows that rel and gen achieve near-parity across setups, especially in 8-shot metric-specific IKE, suggesting models effectively align cross-lingual surface forms. In contrast, loc and port perform substantially worse across datasets and setups. However, port demonstrates greater sensitivity to metric-specific demonstrations than loc. Port benefits from metric-specific demonstrations with more pronounced improvements, especially in datasets like zsRE and WFD.

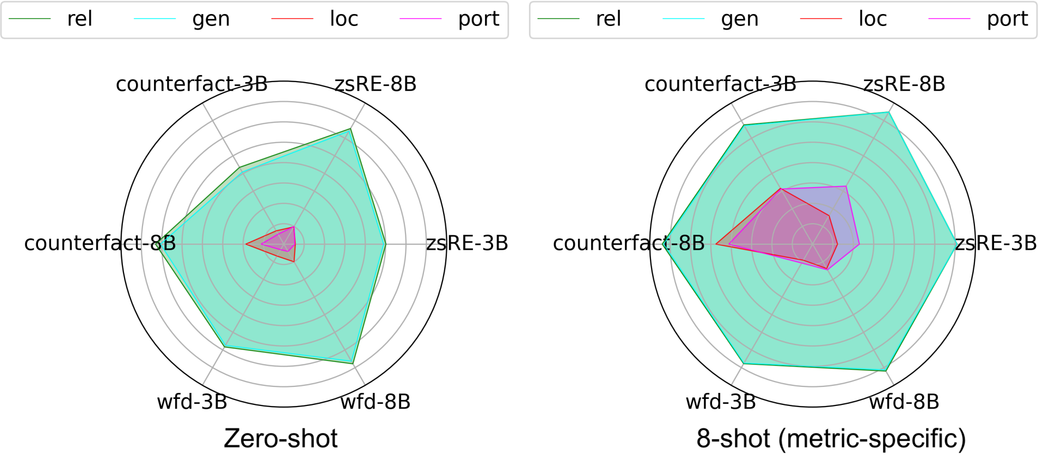


Figure 3.18: Average cross-lingual IKE performance across 52 languages (F1-score).

Impact of Demonstration Strategies The comparison of zero-shot, one-shot, few-shot mixed, and few-shot metric-specific setups reveals the importance of demonstration quality and quantity in cross-lingual IKE. Figure 3.19 illustrates how the performance of `Llama3.1-8B` evolves across setups for each query type and dataset. In the zero-shot setup, models rely solely on their pre-trained capabilities, resulting in limited performance. While one-shot setups provide modest improvements by familiarizing the model with the task format, they fail to address the challenges of loc. A single, randomly selected demonstration often confuses the model, particularly when the demonstration type does not align with the target query type. For loc queries, when encountering rel or gen demonstration types, this misalignment can lead the model to incorrectly repeat the edited knowledge in the target language, further degrading performance. Adding more demonstrations in the few-shot mixed setup yields limited performance gains, particularly for loc queries. However, when demonstrations are tailored to the specific query type being tested, as in the 8-shot metric-specific setup, notable gains are observed. This setup achieves the highest

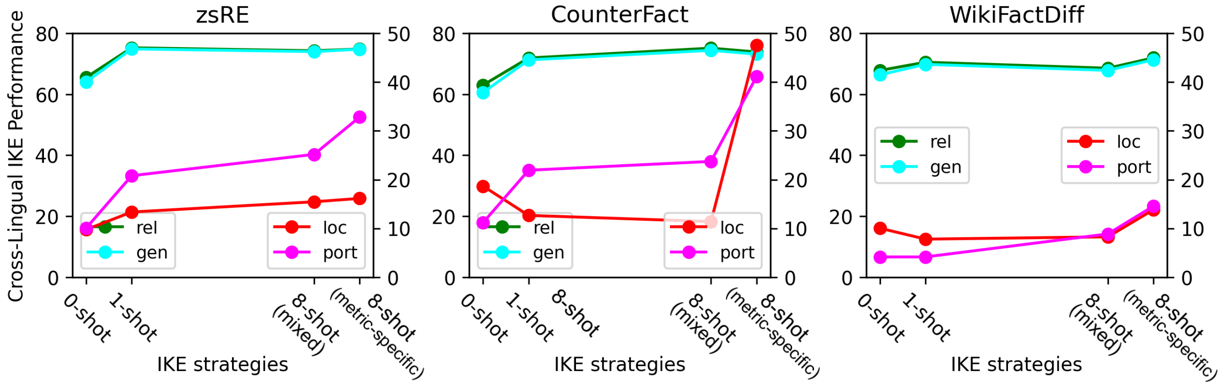


Figure 3.19: Average cross-lingual IKE performance of Llama3.1-8B across 52 languages.

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy
zsRE																										
rel	73	53	66	53	55	26	76	80	79	76	78	81	51	73	78	66	45	76	74	75	75	47	58	75	71	26
gen	73	54	67	53	55	25	76	80	78	76	79	82	51	74	78	67	44	75	74	74	74	46	59	75	71	26
loc	69	19	49	12	27	12	68	63	58	55	75	79	29	71	55	59	19	58	72	45	66	22	34	58	62	16
port	65	29	43	37	42	13	69	54	66	46	72	77	31	69	58	35	29	65	73	40	68	26	32	60	60	13
avg	70	39	56	39	45	19	73	69	70	63	76	80	41	72	67	57	34	68	73	59	70	35	46	67	66	20
CounterFact																										
rel	67	65	52	65	67	54	75	82	60	65	80	89	82	77	35	45	75	37	73	50	79	86	82	50	76	51
gen	72	69	54	69	71	56	79	86	61	68	84	94	85	80	38	50	80	40	74	52	82	90	87	52	80	53
loc	86	50	67	56	58	50	76	69	71	74	91	97	75	94	79	67	60	45	90	51	76	50	68	80	64	34
port	58	47	28	49	53	21	70	70	69	48	73	80	50	79	59	42	35	59	76	39	80	38	37	60	55	16
avg	71	58	50	60	62	45	75	77	65	64	82	90	73	82	53	51	62	45	78	48	79	66	69	60	69	38
WikiFactDiff																										
rel	83	22	76	27	21	14	85	80	79	77	85	81	36	85	80	78	24	76	83	82	77	20	31	79	81	10
gen	83	22	77	26	19	13	83	81	80	77	84	82	35	84	80	77	24	77	82	81	78	20	31	79	81	10
loc	72	40	55	27	46	11	92	92	46	65	56	71	33	93	64	50	23	61	87	53	86	16	38	57	66	12
port	80	32	70	12	32	11	75	34	69	58	77	67	21	84	70	60	27	61	87	60	78	35	22	70	76	9
avg	79	29	70	23	29	12	84	72	68	69	75	75	31	87	74	66	25	69	85	69	79	23	31	71	76	10

	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh
zsRE																										
rel	80	74	71	25	56	59	62	51	80	79	75	73	76	60	78	78	63	56	76	27	54	76	60	42	73	61
gen	79	74	71	24	57	58	63	51	80	78	74	73	75	60	78	77	64	55	77	28	53	76	61	41	74	61
loc	78	68	27	16	22	42	43	30	69	83	56	67	64	41	58	62	53	36	68	7	39	61	35	17	65	19
port	75	73	44	8	32	35	40	35	67	76	66	75	67	54	63	59	42	37	68	5	29	65	52	19	54	48
avg	78	72	53	18	42	48	52	41	74	79	68	72	70	54	69	69	55	46	72	17	44	70	52	30	67	47
CounterFact																										
rel	75	69	92	43	81	50	33	31	79	82	61	71	79	66	58	58	61	64	82	46	80	78	54	60	71	83
gen	79	71	97	45	85	52	33	31	83	86	65	72	82	73	62	62	64	65	86	49	85	83	57	64	74	88
loc	77	93	78	30	76	64	45	40	84	99	79	81	87	78	66	76	70	60	92	31	70	86	71	49	76	84
port	69	78	52	12	38	34	43	37	57	76	74	73	70	70	57	52	51	34	75	13	34	62	57	28	59	54
avg	75	78	80	33	70	50	39	35	76	86	70	74	80	72	61	62	62	56	84	35	67	77	60	50	70	77
WikiFactDiff																										
rel	82	85	23	4	23	77	67	62	91	87	79	77	87	30	78	75	75	59	84	5	56	92	21	16	87	35
gen	82	85	23	5	22	77	65	63	91	86	79	72	87	32	78	75	77	53	83	5	56	92	25	16	86	34
loc	73	88	42	14	13	64	61	75	69	82	45	78	86	36	52	65	61	41	44	6	39	61	38	46	60	52
port	70	79	14	3	16	34	37	32	78	86	71	91	85	46	70	79	74	15	82	2	42	79	39	16	77	30
avg	77	84	26	7	19	63	57	58	82	85	69	80	86	36	70	74	72	42	73	4	48	81	31	23	77	38

Table 3.33: Cross-lingual IKE performance of Llama3.1-8B in 52 languages under the 8-shot metric-specific setup.

scores for loc and portability, as shown in Figure 3.19. These findings underscore the importance of demonstration quality and specificity in maximizing the benefits of in-context learning for cross-lingual knowledge editing.

3.4.6 Language Performance Variance in Cross-Lingual IKE

To analyze cross-lingual performance variation, we use a normalized metric that uses the exact match (EM) of English as a reference to highlight cross-lingual transfer differences. Specifically, we calculate the ratio of each target language's EM performance to English. As is visually evident in Table 3.33, some languages perform particularly poorly, while others achieve results closer to English, motivating further investigation into the factors influencing these differences.

Correlation Between Language Properties and Performance We conducted a correlation analysis between language properties (derived using Lang2Vec (Littell et al., 2017), details in Appendix) and query-type performance across datasets. As revealed in Figure 3.20, syntactic, phonological, and geographic similarities with English positively correlate with performance, particularly for loc and port queries However, two notable exceptions emerge: rel and gen queries in CounterFact show no significant correlation with language properties, likely because these query types achieve uniformly high performance across all languages. Additionally, genetic similarity (language family) shows no meaningful correlation across datasets.

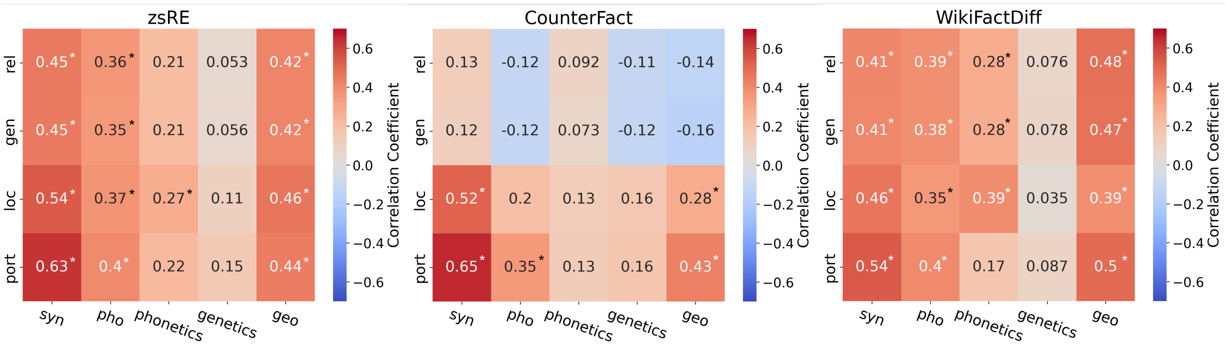
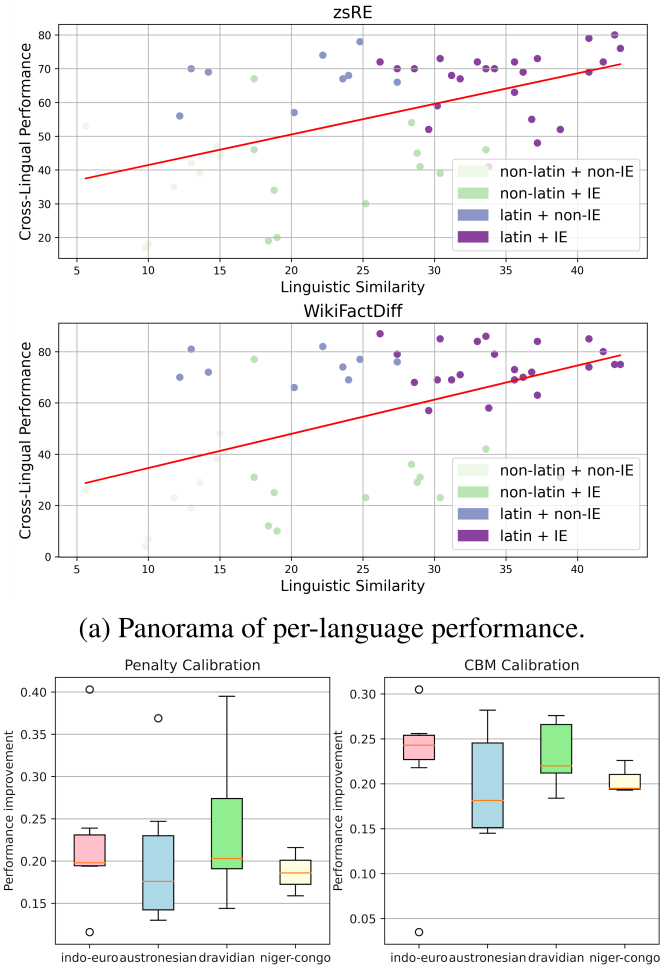


Figure 3.20: Correlation between linguistic properties and IKE performance of 52 languages. Experimental results of Llama3.1-8B under the 8-shot metric-specific setup. Significant correlations are marked with $*$ ($p < 0.05$).

Impact of Script and Language Family To further explore the role of script and language family, we grouped languages into four clusters based on script type (Latin vs. non-Latin) and language family (Indo-European vs. non-Indo-European). Figure 3.21(a) reveals that script type plays a more critical role than language family. Non-Latin languages, regardless of their family, perform worse than Latin-script languages. This trend is consistent across datasets and query types, as further supported by Figure 3.21(b). For example, the non-Latin + non-IE and non-Latin + IE groups exhibit similar performance, both significantly lower than the Latin + IE and Latin + non-IE groups.

Language Confusion and Script Effects A qualitative error analysis highlights language confusion as a key factor explaining why script type matters. Language confusion occurs when the model generates answers in English instead of the target language (Marchisio et al., 2024), even when explicitly instructed to use the target language. This issue is particularly prevalent in code-switched prompts and disproportionately affects non-Latin languages. For example, while the model may produce the correct answer in English, it is considered a failure if the output is not in the target language. Table 3.34 highlights instances of language confusion that occur when queries are presented in Chinese. This problem is exacerbated for non-Latin languages, as their distinct scripts reduce the likelihood of overlapping writing forms between English and the target language, further degrading performance.



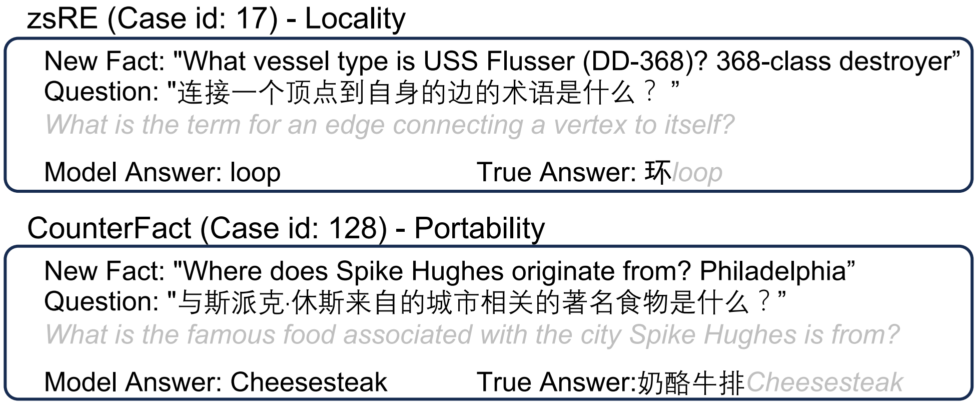
(a) Panorama of per-language performance.

(b) IE: Indo-European language family.

Figure 3.21: Impact of Script and Language Family.

3.4.7 Sum-Up

We present BMIKE-53, a multilingual KE benchmark, and leverage it to investigate the potential of gradient-free in-context learning methods for cross-lingual knowledge editing. Our experiments demonstrate that tailored demonstration strategies significantly enhance KE performance, with metric-specific demonstrations improving locality and portability. Additionally, linguistic properties, particularly script type, strongly influence cross-lingual knowledge transfer. We hope that BMIKE-53 will inspire further research in multilingual KE, advancing the understanding and capabilities of LLMs across diverse linguistic contexts.



zsRE (Case id: 17) - Locality

New Fact: "What vessel type is USS Flusser (DD-368)? 368-class destroyer”
Question: "连接一个顶点到自身的边的术语是什么？ ”
What is the term for an edge connecting a vertex to itself?

Model Answer: loop True Answer: 环*loop*

CounterFact (Case id: 128) - Portability

New Fact: "Where does Spike Hughes originate from? Philadelphia”
Question: "与斯派克·休斯来自的城市相关的著名食物是什么？”
What is the famous food associated with the city Spike Hughes is from?

Model Answer: Cheesesteak True Answer:奶酪牛排*Cheesesteak*

Table 3.34: Examples of language confusion from the 8-shot metric-specific setup with Llama3.1-8B.

Chapter 4

Prompt-Based Fine-Tuning for Zero-Shot Cross-Lingual Transfer

Summary of This Chapter

Building on the foundations of prompt-based learning established in the previous chapter (Chapter 3), this chapter advances the exploration of prompt-based methods by focusing on **prompt-based fine-tuning** within the paradigm of zero-shot cross-lingual transfer. Zero-shot cross-lingual transfer is a powerful approach in multilingual NLP, where a model is fine-tuned on labeled data from a single source language, typically English, and then directly evaluated on target language samples without any further adaptation or labeled data in the target language. This paradigm is especially valuable for low-resource and underrepresented languages, where annotated data is scarce or unavailable, and it offers a scalable solution for extending language technologies to a broader linguistic landscape.

Despite the promise of prompt-based learning, its application to fine-tuning for multilingual NLP, particularly in zero-shot cross-lingual transfer, remains underexplored. Most prior work has focused on prompt-based inference or in-context learning, leaving open questions about the comparative advantages of prompt-based fine-tuning over traditional (vanilla) fine-tuning approaches. To address this gap, the first part of this chapter presents a comprehensive empirical investigation into prompt-based fine-tuning for cross-lingual language understanding. Through the PROFIT pipeline, we systematically compare prompt-based and vanilla fine-tuning across a range of multilingual classification tasks, including sentiment analysis, paraphrase identification, and natural language inference. Our findings reveal that prompt-based fine-tuning not only consistently outperforms vanilla fine-tuning in full-data scenarios, but also exhibits even greater advantages in few-shot settings. Furthermore, we analyze how factors such as language similarity and pretraining data size influence the effectiveness of cross-lingual transfer, providing new insights into the dynamics of multilingual adaptation (§4.1).

Beyond sentence-level classification, the chapter extends prompt-based fine-tuning to structured prediction tasks, which are critical for many real-world NLP applications. We introduce the ToPro methodology, a token-level prompt decomposition approach for sequence labeling tasks such as part-of-speech (POS) tagging and named entity recognition (NER). By decomposing the input into token-level prompts, ToPro enables more granular and robust learning, particularly benefiting languages that are typologically distant from English. Our experiments demonstrate that ToPro-based fine-tuning achieves state-of-the-art zero-shot cross-lingual performance, outperforming both vanilla fine-tuning and prompt-tuning baselines (§4.2).

Finally, this chapter explores the application of zero-shot cross-lingual transfer to historical language processing, a domain where annotated resources are especially scarce. We develop a delexicalized constituency parser for Middle High German (MHG), leveraging the structural continuity between MHG and Modern German. By fine-tuning on modern German treebanks and transferring to MHG without any annotated MHG training data, our approach demonstrates the feasibility and effectiveness of cross-lingual transfer for syntactic parsing in ancient languages, providing a new tool for historical language processing (§4.3).

In summary, this chapter demonstrates that prompt-based fine-tuning is a versatile and powerful tool for zero-shot cross-lingual transfer, enabling robust adaptation across both classification and structured prediction tasks.

4.1 Prompt-Based Fine-Tuning vs. Vanilla Fine-Tuning for Cross-Lingual Language Understanding

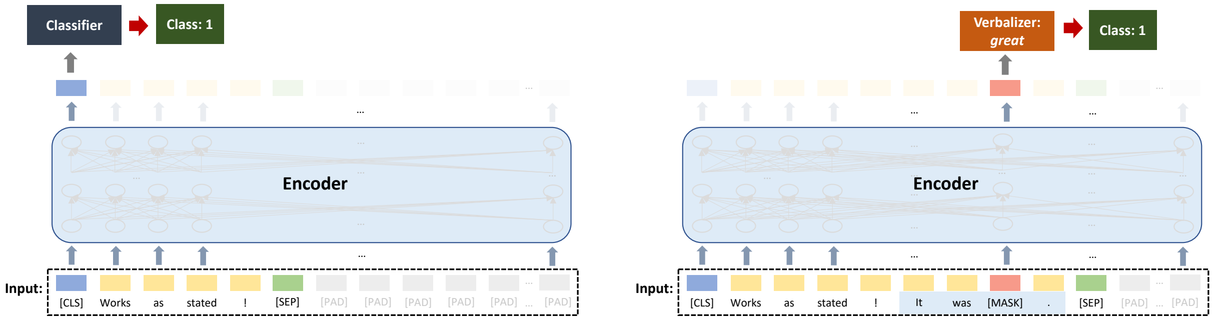
This section corresponds to the following work:

> Bolei Ma*, **Ercong Nie***, Helmut Schmid, and Hinrich Schütze. 2023. Is Prompt-Based fine-tuning Always Better than Vanilla fine-tuning? Insights from Cross-Lingual Language Understanding. In Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023), pages 1–16, Ingolstadt, Germany. Association for Computational Linguistics.
> * equal contributions.

Declaration of Co-Authorship. I proposed the research question and designed the research framework for conducting a comparative empirical study between prompt-based fine-tuning and vanilla fine-tuning for zero-shot cross-lingual transfer. I wrote the code framework. Bolei Ma ran the experiments and conducted the analysis. Bolei Ma and I completed the paper writing together. Helmut Schmid and Hinrich Schütze supervised the project and provided valuable feedback.

Summary of This Section

Multilingual pretrained language models (MPLMs) have demonstrated substantial performance improvements in zero-shot cross-lingual transfer across various natural language understanding tasks by fine-tuning MPLMs on task-specific labelled data of a source language (e.g., English) and evaluating on a wide range of target languages. Recent studies show that prompt-based fine-tuning surpasses regular fine-tuning in few-shot scenarios. However, the exploration of prompt-based learning in multilingual tasks remains limited. In this study, we propose the **ProFiT** pipeline to investigate the cross-lingual capabilities of **Pro**mpt-based **Fi**ne-**T**uning. We conduct comprehensive experiments on diverse cross-lingual language understanding tasks (sentiment classification, paraphrase identification, and natural language inference) and empirically analyze the variation trends of prompt-based fine-tuning performance in cross-lingual transfer across different few-shot and full-data settings. Our results reveal the effectiveness and versatility of prompt-based fine-tuning in cross-lingual language understanding. Our findings indicate that prompt-based fine-tuning outperforms vanilla fine-tuning in full-data scenarios and exhibits greater advantages in few-shot scenarios, with different performance patterns dependent on task types. Additionally, we analyze underlying factors such as language similarity and pretraining data size that impact the cross-lingual performance of prompt-based fine-tuning. Overall, our work provides valuable insights into the cross-lingual prowess of prompt-based fine-tuning.



(a) Vanilla fine-tuning

(b) Language Diversity

Figure 4.1: The comparison of vanilla fine-tuning and prompt-based fine-tuning. [CLS], [SEP], [MASK], and [PAD] are special tokens in the encoder vocabulary. The verbalizer is a function mapping from the task label set to a subset of the encoder vocabulary. Input tokens in blue represent the prompt pattern.

4.1.1 Background and Research Questions

Pretrained language models (PLMs) (Devlin et al., 2019; Yang et al., 2019b; Radford et al., 2019), trained on massive amounts of unlabelled data in a self-supervised manner, have shown strong performance after fine-tuning on task-specific labelled data for a given downstream task, such as sentence classification (Zhuang et al., 2021), text summarization (Zhang et al., 2020a), or dialogue generation Liu et al. (2023c). *Prompt-based learning* (Brown et al., 2020; Schick and Schütze, 2021a,b,c) has recently emerged as a notable advancement, surpassing regular fine-tuning approaches in few-shot scenarios Liu et al. (2023a). In prompt-based learning, as introduced in the previous chapter (Chapter 3), downstream tasks are reformulated to resemble the types of problems tackled during the PLM's original pretraining by using a textual prompt. For example, in Figure 4.1, an input sentence of the binary sentiment analysis task "`Works as stated!`" can be reformulated with a prompt pattern $P(X) = X \circ$ "`It was [MASK].`" as "`Works as stated! It was [MASK].`" where $\circ$ is the string concatenation operator. We use a *verbalizer* which maps the class label to a *label word*. In this example, the class labels POSITIVE and NEGATIVE can be verbalized as "`great`" and "`bad`". By comparing the probabilities of the label words "`great`" and "`bad`" as fillers of the `[MASK]` token, we can predict the correct class label. In the example above, a natural language understanding (NLU) task is transformed into a masked language modeling (MLM) problem, which is the same as the PLM's pretraining objective.

In the work of the previous chapter (Chapter 3), we used prompt-based learning for model inference and prediction, i.e., without training the model parameters. In fact, the reformulated input can also be used for fine-tuning, i.e. *prompt-based fine-tuning*, which is the research focus of this chapter. Figure 4.1 shows the difference between prompt-based fine-tuning and vanilla fine-tuning. Vanilla fine-tuning solely relies on the hidden embedding of the `[CLS]` token. In contrast, prompt-based fine-tuning makes use of both the semantic information from the task labels and the prior knowledge encoded in the pretraining phase. Recent empirical studies of few-shot learning showed advantages of prompt-based fine-tuning over vanilla fine-tuning (Gao et al., 2021; Li and Liang, 2021).

When applied to multilingual pretrained language models (MPLMs), prompt-based fine-tuning also enables zero-shot[1] cross-lingual transfer. MPLMs such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) are pretrained on huge multilingual corpora and show strong multilinguality (Pires et al., 2019; Dufter and Schütze, 2020; Liang et al., 2021). They have become the dominant paradigm for zero-shot cross-lingual transfer, where annotated training data is available for some source language (e.g. English) but not for the target language (Wu and Dredze, 2019; Hu et al., 2020b). Zhao and Schütze (2021) proposed prompt-based fine-tuning for cross-lingual transfer. Their work focused on few-shot fine-tuning. Their experimental results for the natural language inference task showed that prompt-based fine-tuning performed better in few-shot cross-lingual transfer than vanilla fine-tuning. However, prior studies failed to examine whether prompt-based learning is also advantageous when training data is not scarce. Therefore, we conduct a comprehensive investigation on diverse cross-lingual language understanding tasks in both full-data and few-shot settings in order to shed more light on the cross-lingual capabilities of prompt-based fine-tuning.

In contrast to most previous research on prompting, our work is not restricted to monolingual or few-shot scenarios. Instead, we explore a wide range of few-shot settings. We adopt a

[1] In this section, "zero-shot" in "zero-shot cross-lingual transfer" refers to the number of target language training data, i.e., no target language data is provided, while "few-shot" in "few-shot fine-tuning" refers to the source language used for fine-tuning, i.e., a few source language data is provided for the fine-tuning of the MPLM. The fine-tuned model is then zero-shot transferred to the target language.

multilingual perspective and aim to uncover the nuances of performance variations associated with prompt-based fine-tuning. To this end, we implement the PROFIT pipeline and carry out an extensive set of experiments encompassing three representative cross-lingual language understanding tasks: sentiment analysis (Amazon Reviews), paraphrase identification (PAWS-X), and natural language inference (XNLI). Our task selection covers single-sentence classification, sentence pair classification, and inference tasks, considering both binary and multi-fold classifications. Our work provides insights into the effectiveness and versatility of prompt-based fine-tuning in cross-lingual language understanding.

Research Questions In this section, we analyze how the performance of prompt-based fine-tuning varies with the size of the labelled source language data for zero-shot cross-lingual transfer tasks. We examine a wide range of factors that could have an impact on cross-lingual transfer performance. We attempt to address the following pivotal research questions:

RQ1 *Does prompt-based fine-tuning outperform vanilla fine-tuning in the full-data scenario in different NLU tasks?*
We propose the PROFIT pipeline for systematically conducting the cross-lingual transfer experiments. We carry out zero-shot cross-lingual transfer experiments on three different NLU tasks using all the available English training data. By comparing the results of vanilla fine-tuning and PROFIT for different MPLMs, we find that in the full-data scenario, PROFIT still achieves better cross-lingual performance than vanilla fine-tuning.

RQ2 *Is prompt-based fine-tuning always better than vanilla fine-tuning?*
We investigate how the cross-lingual performance depends on the size of the English training data. Our findings substantiate that the PROFIT exhibits greater advantages in few-shot scenarios compared to full-data scenarios. The specific patterns of performance change are contingent upon the task types.

RQ3 *What underlying factors could affect the cross-lingual performance of* PROFIT *?*
We extensively analyze the factors that could influence the cross-lingual performance of PROFIT , encompassing language similarity, pretraining data size of target languages, etc.

4.1.2 Methodology

The purpose of this study is to improve upon the cross-lingual transfer performance of vanilla fine-tuning. In vanilla settings of zero-shot cross-lingual transfer, the MPLM is directly fine-tuned with training data in a source language (English). The fine-tuned model is then applied to predict the test data in target languages.

In prompt-based learning, we need a pattern-verbalizer pair (PVP) (Schick and Schütze, 2021a) consisting of (i) a *prompt pattern* which converts the input text into a cloze-style question with a mask token, and (ii) a representative word (called *verbalizer*) for each possible

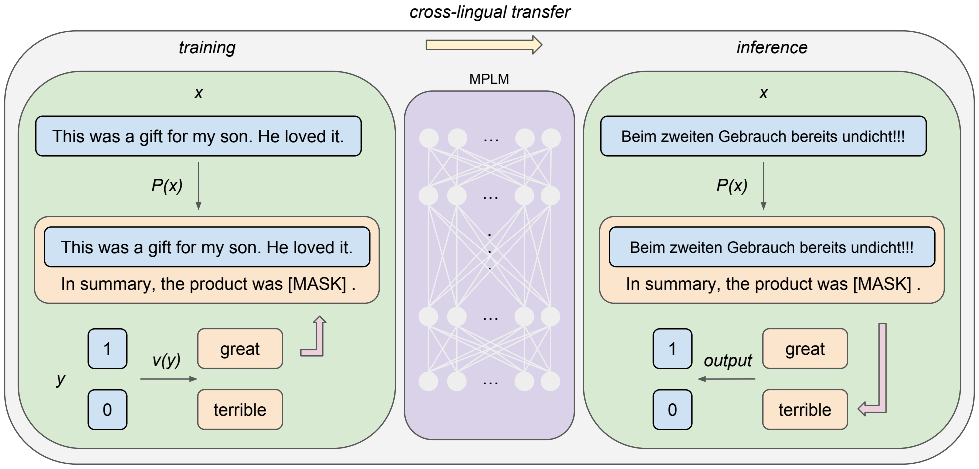

cross-lingual transfer
training
inference
MPLM
x
This was a gift for my son. He loved it.
P(x)
This was a gift for my son. He loved it.
In summary, the product was [MASK] .
1
v(y)
great
y
0
terrible
x
Beim zweiten Gebrauch bereits undicht!!!
P(x)
Beim zweiten Gebrauch bereits undicht!!!
In summary, the product was [MASK] .
1
output
great
0
terrible

Figure 4.2: PROFIT pipeline of training and cross-lingual transfer with examples. X is an input sentence, and $P(X)$ denotes the prompt pattern that reformulates the input into a prompt. $v(y)$ is the verbalizer which maps each class label y onto a word from the source language vocabulary.

class. In our PROFIT approach, a PVP is combined with training data in English during fine-tuning. As the *training* block in Figure 4.2 shows, a prompt pattern such as $P(X) = X \circ$ "`In summary, the product was [MASK].`" is filled with an input example X "`This was a gift for my son. He loved it.`" A verbalizer such as {0 → "terrible", 1 → "great"} is used to map the original labels {0,1} onto words. The MPLM takes the filled pattern "`This was a gift for my son. He loved it. In summary, the product was [MASK].`", as input and returns for each of the two verbalizers "terrible" and "great" its probability of being the masked token. Thus, it uses the PVP to reformulate the sentence classification task of vanilla fine-tuning into a masked token prediction task.

More formally, let D=$\{(X_1, y_1), \dots, (X_n, y_n)\}$ denote the set of training examples in the source language, where X_1, ..., X_n are text samples and y_1, ..., y_n are class labels from a label set Y. The prompt pattern $P(.)$ transforms an input sentence X into a cloze-style question with a masked token. The pretrained language model M with trainable parameters θ performs masked token prediction and returns the probabilities $p = M(P(X), \theta)$ of all candidate words for the masked token in $P(X)$. The verbalizer $v(.)$ is a bijective mapping from the set of class labels Y to a set of verbalised words V from the source language vocabulary. We predict the class $\hat{y}$ whose verbalizer $v(\hat{y})$ received the highest probability from model M:

$$\hat{y} = \arg\max_{y \in Y} p(v(y)) \tag{4.1}$$

We fine-tune the parameters θ of model M by minimizing the cross-entropy loss function ℓ on D:

$$\hat{\theta} = \arg\min_{\theta} \sum_{(X,y) \in D} \ell(v(y), M(P(X), \theta)) \tag{4.2}$$

The model with the fine-tuned parameters $\hat{\theta}$ is used to predict the class labels of the target language examples $D' = \{X'_1, \dots, X'_n\}$ using the same prompt pattern and verbalizer as during fine-tuning (see *inference* block in Figure 4.2). The best label y'_i for each example X'_i is predicted according to Eq. 4.1.

In contrast to vanilla fine-tuning, prompt-based methods such as PROFIT only transform the training data with the prompt pattern P and the verbalizer v, but leave the model architecture unchanged, thus not hindering the efficiency of Vanilla much (Shi and Lipani, 2023). No extra parameters have to be trained from scratch. By reformulating the sentence classification task into a masked token prediction (MTP) task, we can better take advantage of the knowledge that the model has acquired during MTP pretraining.

In the cross-lingual setting, we simply apply the same functions P and v to the target language examples without further modifications.

4.1.3 Experimental Setups

Datasets In order to investigate the performance on diverse NLU tasks, three representative different classification tasks on NLU are selected for evaluation in this work: sentiment analysis on Amazon product reviews (Keung et al., 2020), paraphrase identification on PAWS-X (Yang et al., 2019a), and natural language inference on XNLI (Conneau et al., 2018).

Amazon Reviews Dataset (Keung et al., 2020) contains product reviews with 5-star ratings from 1 to 5. The multilingual version of this dataset consists of test data in English and 5 other languages. We use the following prompt pattern $P(X)$ and verbalizer $v(y)$ for each review example (X, y):

- $P(X) = X \circ$ "All in all, it was [MASK]."
- $v(1) =$ "terrible", $v(2) =$ "bad",
  $v(3) =$ "ok", $v(4) =$"good", $v(5) =$ "great"

PAWS-X is a multilingual version of PAWS (Zhang et al., 2019a), which consists of challenging paraphrase identification pairs from Wikipedia and Quora. Each data item comprises two sentences. The task is to predict whether the two sentences are paraphrases. The labels are binary: 1 for paraphrase, 0 for non-paraphrase. PAWS-X consists of datasets in English and 6 other languages. For a given sentence pair X_1 and X_2, we design the pattern and verbalizer as:

- $P(X_1, X_2) = X_1 \circ$"? [MASK], " $\circ X_2$
- $v(0) =$ "Wrong", $v(1) =$ "Right"

XNLI is a multilingual version of the MultiNLI dataset (Williams et al., 2018). The text in each data item consists of two sentences. Sentence A is the premise, and sentence B is the hypothesis. The task is to predict the type of inference between the given premise and hypothesis among the three types: "entailment" (0), "neutral" (1), and "contradiction" (2). It is a kind of multi-class natural language inference task. XNLI consists of datasets in English and 14 other languages. For a given sentence pair X_1 and X_2, we design the pattern and verbalizer as:

- $P(X_1, X_2) = X_1 \circ$ "? [MASK], " $\circ X_2$

- $v(0)$ = "Yes", $v(1)$ = "Maybe", $v(2)$ = "No"

In Table 4.1, we show a basic statistic view of the Amazon Review (Keung et al., 2020), PAWS-X (Zhang et al., 2019a), and XNLI (Williams et al., 2018) datasets. We use the original train-dev-test split from the datasets. For training and validation, we use the English train and dev dataset, and for test we use the test sets of all languages. The test data size for each target language is the same in all tasks.

Task	Size			#Labels
	— Train —	— Dev —	— Test —	
Amazon	200 000	5 000	5 000	5
PAWS-X	49 401	2 000	2 000	2
XNLI	392 702	2 490	5 010	3

Table 4.1: Overview of the three datasets. Train and dev data size refers to the number of samples for English. Test data size refers to the number of samples for each target language.

Baseline The following baselines are considered and compared to our ProFiT approach:

MAJ. The majority baseline. It always assigns the majority class from the training data.

Direct. The pattern filled with the input sample is directly fed to the MPLM for prediction, without fine-tuning. This is the zero-shot scenario.

Vanilla. The standard fine-tuning method which predicts the class from the hidden embedding of the `[CLS]` token without using a prompt pattern. We use the cross-entropy loss as the objective function for fine-tuning and AdamW for optimization with a learning rate of 1e-5 and 5 training epochs. The fine-tuned models are then used to predict the test data.

Multilingual Models To solve the classification tasks with cross-lingual transfer, we use the pretrained multilingual BERT model (Devlin et al., 2019) "`bert-base-multilingual-cased`" (M) and the XLM-R model (Conneau et al., 2020) "`xlm-roberta-base`" (X) from the Huggingface Transformers library (Wolf et al., 2020). Both models are evaluated with the methods Vanilla and ProFiT . During training, we used the same hyperparameters for Vanilla and ProFiT to keep the variables consistent for comparison. The chosen hyperparameters for both full-shot training and few-shot training are documented in Table 4.2. To avoid random effects on training, we trained each experiment with 5 different random seeds $\{10, 42, 421, 510, 1218\}$ and took the average results.

4.1.4 Results

Hyperparameter	Full	Few-shot
EPOCHS	5	50
LEARNING_RATE	1e-5	1e-5
BATCH_SIZE	8	1
GRADIENT_ACCUMULATION_STEPS	4	2
MAX_SEQ_LENGTH	128	128
EARLY_STOPPING_PATIENCE	-	3

Table 4.2: Hyperparameters

	Amazon	PAWS-X	XNLI	Avg.
MAJ	20	55.81	33.33	36.17
Direct-mBERT	20.21	45.05	35.05	33.44
Vanilla-mBERT	42.97	80.24	65.05	62.75
ProFiT -mBERT	**43.98**	**82.16**	**65.79**	**63.98**
Direct-XLM-R	21.98	51.10	35.68	36.25
Vanilla-XLM-R	54.56	82.51	73.61	70.22
ProFiT -XLM-R	**54.66**	**82.73**	**73.82**	**70.40**

Table 4.3: Overview of results

Main Results Table 4.3 gives an overview of the experimental results. ProFiT outperforms the MAJ baseline with both mBERT and XLM-R for all three classification tasks. ProFiT also outperforms the Direct and Vanilla baselines in both mBERT and XLM-R settings: When trained with mBERT, the performance is improved by **23.77%**, **37.11%** and **30.74%** compared to Direct on Amazon, PAWS-X and XNLI respectively, and by **1.01%**, **1.92%** and **0.74%** compared to Vanilla. When trained with XLM-R, the performance is improved by **32.68%**, **31.63%** and **38.14%** compared to Direct, and by **0.10%**, **0.22%** and **0.21%** compared to Vanilla respectively.

While ProFiT outperforms all baselines on all three tasks, the degree of improvement differs. The improvements of ProFiT over Vanilla when trained with mBERT (**+1.23%**) are larger than the improvements when trained with XLM-R (**+0.18%**).

We further conducted T-tests for the results of Vanilla and ProFiT with different random seeds. Table 4.5 shows the T-test results with p values for each task with mBERT and XLM-R models. We can see that the p values of all three tasks with the mBERT model are under 0.05, indicating that the performance gain of ProFiT is significant with mBERT, while the p values of all three tasks with the XLM-R model are bigger than 0.05, showing no significant performance difference.

One reason for the performance difference of the two models could be that the XLM-R model was pretrained on far more data than mBERT and is also much bigger, so that the Vanilla per-

Task	Model	en	ar	bg	de	el	es	fr	hi	ja	ko	ru	sw	th	tr	ur	vi	zh	avg.
Amazon	Vanilla-M	58.92	-	-	45.69	-	48.02	47.45	-	35.07	-	-	-	-	-	-	-	**38.63**	42.97
	PROFIT -M	**59.05**	-	-	**46.66**	-	**49.30**	**48.38**	-	**37.31**	-	-	-	-	-	-	-	38.26	**43.98**
	Vanilla-X	59.61	-	-	**60.14**	-	55.24	55.66	-	51.93	-	-	-	-	-	-	-	49.82	54.56
	PROFIT -X	**60.06**	-	-	59.60	-	**55.72**	**55.89**	-	**52.34**	-	-	-	-	-	-	-	**49.75**	**54.66**
PAWS-X	Vanilla-M	93.85	-	-	84.94	-	87.11	86.55	-	73.39	72.44	-	-	-	-	-	-	77.01	80.24
	PROFIT -M	**94.21**	-	-	**86.06**	-	**88.17**	**87.91**	-	**75.79**	**75.82**	-	-	-	-	-	-	**79.22**	**82.16**
	Vanilla-X	94.33	-	-	86.92	-	88.55	89.04	-	**76.07**	74.71	-	-	-	-	-	-	79.75	82.51
	PROFIT -X	**94.90**	-	-	**87.06**	-	**88.87**	**88.86**	-	75.53	**75.40**	-	-	-	-	-	-	**80.63**	**82.73**
XNLI	Vanilla-M	82.57	65.12	68.97	71.40	66.30	74.22	73.68	60.02	-	-	68.95	50.24	53.15	62.02	57.96	69.80	68.91	65.05
	PROFIT -M	82.57	**65.55**	**69.47**	**71.57**	**67.43**	**75.10**	**74.57**	**60.57**	-	-	**69.55**	**51.13**	**54.58**	**62.64**	**58.04**	**70.74**	**70.08**	**65.79**
	Vanilla-X	84.91	**71.86**	77.78	76.86	75.96	79.25	78.21	69.92	-	-	**75.79**	**65.21**	72.02	73.12	66.07	74.71	73.72	73.61
	PROFIT -X	**84.97**	71.81	**77.92**	**77.35**	**76.11**	**79.31**	**78.75**	**70.10**	-	-	75.43	65.13	**72.39**	**73.23**	**66.95**	**75.05**	**73.92**	**73.82**

Table 4.4: Detailed cross-lingual performance results on three classification tasks. When calculating the average (avg.), due to the aim of zero-shot cross-lingual transfer, the performance results of the source language (English) are not taken into account. Model M stands for mBERT, and X for XLM-R.

Model	Amazon	PAWS-X	XNLI
mBERT	0.005	0.003	0.005
XLM-R	0.40*	0.46*	0.44*

Table 4.5: T-Test results (p) for results of Vanilla and PROFIT with different random seeds. Insignificant results with a p value > 0.05 are marked with *.

formance with XLM-R fine-tuning is much better than with mBERT in cross-lingual context (Conneau et al., 2020; Lauscher et al., 2020), leaving less space for improvement.

A detailed overview of the cross-lingual performance of PROFIT compared to Vanilla for each target language is presented in Table 4.4. Although the overall performance of PROFIT is better than Vanilla for all three tasks in both mBERT and XLM-R settings, individual differences between languages can be noticed. On Amazon, with mBERT, the improvement in Japanese (ja) (**+2.24%**) is far greater than on average, whereas Chinese (zh) shows no improvement (**-0.37%**); with XLM-R, PROFIT performs slightly worse than Vanilla on both Chinese with **-0.07%** and German (de) with **-0.54%**. On PAWS-X, Korean (ko) shows a larger improvement (**+3.38%**) than average with mBERT, and with XLM-R, whereas French (fr) (**-0.18%**) and Japanese (**-0.54%**) show a slightly worse performance than Vanilla. On XNLI, we find improvements for all languages with mBERT, and with XLM-R, Arabic (ar) (**-0.06%**), Russian (ru) (**-0.36%**), and Swahili (sw) (**-0.08%**) show slightly worse performance than Vanilla.

We conclude that the performance gain of PROFIT over Vanilla depends on the models and languages. In §4.1.5, we will further investigate how linguistic factors influence cross-lingual transfer performance.

Few-shot Ablations Previous studies show that the prompt framework is more effective than fine-tuning when training data is scarce (Zhao and Schütze, 2021; Qi et al., 2022). We investigated how the performance changes as the number of training samples K increases in few-shot settings. The training and validation data are randomly sampled with $K \in \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$ shots per class from the English training data.

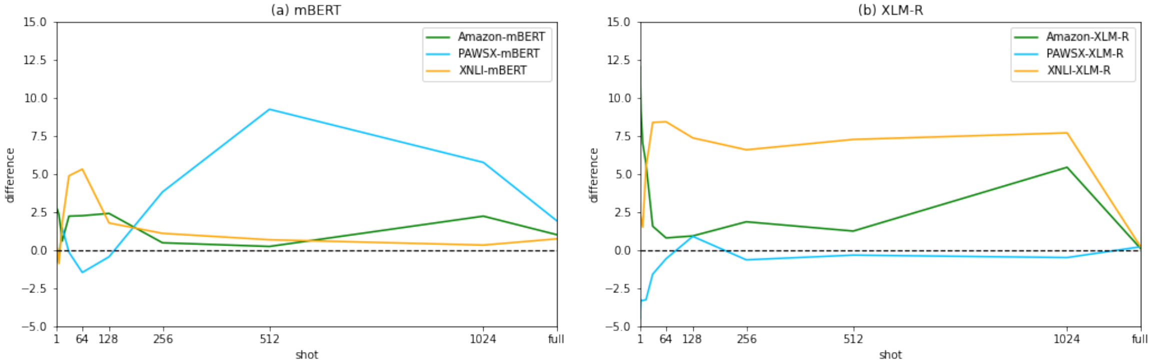


Figure 4.3: Performance difference between ProFiT and Vanilla in different few-shot settings and full training setting on three NLU tasks with both mBERT and XLM-R models.

The detailed results of few-shot ablations can be found in Table 13, Table 14, and Table 15 in Appendix D. Figure 4.3 shows the performance changes on all three tasks with both mBERT and XLM-R models. On the Amazon task, the performance improvement for smaller numbers of shots is greater than for full training. As the number of shots increases, the improvement decreases accordingly. This implies that on the sentiment analysis task, ProFiT is most valuable with small training data. On XNLI, the improvement of ProFiT over Vanilla is first small within small shots. It then gets greater, as K increases, and drops again, as a bigger K towards the full data size shows up. We conclude that on NLI tasks such as XNLI, ProFiT is most effective in few-shot settings with a certain number of K. On PAWS-X, no obvious difference in few-shot settings can be found with mBERT in small shots, but in bigger shots, there is greater improvement with $K \in \{256, 512, 1024\}$; however, with XLM-R, ProFiT shows almost no performance improvement over Vanilla.

Overall, sentiment analysis exhibits a clearer performance improvement for smaller numbers of shots, whereas the language inference and paraphrase tasks show greater performance enhancements in few-shot scenarios with larger K. This might be due to difficulties with pairwise inputs in these tasks, where we aim to identify the relationship between a pair of sentences. When it comes to transferring knowledge of sentence relationships, more examples are needed for successful learning than in sentiment analysis tasks, where semantic information from comparable cross-lingual sentences can be directly transferred.

4.1.5 Cross-Lingual Analysis

In previous empirical studies of cross-lingual transfer learning (Lauscher et al., 2020; Nie et al., 2023a), several key factors were identified to exert a great effect on the cross-lingual perfor-

lang	Typological & Phylogenetic Sim.						Lexical Sim.			Size	Task Performance					
	SYN	PHO	INV	FAM	GEO	**Sim$_1$**	*UMAP*	*SVD*	**Sim$_2$**		amazon-M	amazon-X	pawsx-M	pawsx-X	xnli-M	xnli-X
ar	65.47	70.06	75.88	0.00	97.04	**61.69**	-1.90	4.87	**1.49**	**20.20**	-	-	-	-	65.55	71.81
bg	78.78	90.45	70.02	13.61	99.01	**70.38**	8.65	33.21	**20.93**	**18.15**	-	-	-	-	69.47	77.92
de	79.05	83.62	77.62	54.43	99.76	**78.90**	83.42	76.83	**80.13**	**21.42**	46.66	59.60	86.06	87.06	71.57	77.35
el	73.19	95.35	64.75	14.91	98.95	**69.43**	1.24	24.81	**13.03**	**17.76**	-	-	-	-	67.43	76.11
es	84.97	85.81	64.99	9.62	99.59	**69.00**	1.61	28.30	**14.96**	**20.83**	49.30	55.72	88.17	88.87	75.10	79.31
fr	76.83	75.26	73.64	9.62	99.93	**67.06**	1.34	31.76	**16.55**	**21.27**	48.38	55.89	87.91	88.86	74.57	78.75
hi	58.79	85.81	76.53	12.60	91.10	**64.97**	1.20	21.11	**11.16**	**17.26**	-	-	-	-	60.57	70.10
ja	49.63	64.44	65.92	0.00	85.65	**53.13**	-	-	-	**20.39**	37.31	52.34	75.79	75.53	-	-
ko	55.66	74.62	71.04	0.00	86.93	**57.65**	-0.22	12.42	**6.10**	**19.28**	-	-	75.82	75.40	-	-
ru	75.74	90.45	63.17	16.67	95.81	**68.37**	8.63	32.60	**20.62**	**20.87**	-	-	-	-	69.55	75.43
sw	42.26	90.91	76.16	0.00	91.50	**60.17**	-9.05	-7.18	**-8.12**	**16.23**	-	-	-	-	51.13	65.13
th	65.20	81.82	78.88	0.00	85.25	**62.23**	-0.21	3.82	**1.81**	**17.25**	-	-	-	-	54.58	72.39
tr	43.36	85.81	68.49	0.00	98.25	**59.18**	-7.80	-1.56	**-4.68**	**19.00**	-	-	-	-	62.64	73.23
ur	50.01	0.00	71.56	12.60	92.54	**45.34**	1.35	24.92	**13.14**	**17.54**	-	-	-	-	58.04	66.95
vi	64.92	78.33	74.76	0.00	85.25	**60.65**	0.86	-18.50	**-8.82**	**20.29**	-	-	-	-	70.74	75.05
zh	73.49	78.33	74.91	0.00	88.42	**63.03**	-	-	-	**20.37**	38.26	49.75	79.22	80.63	70.08	73.92

Table 4.6: Overview of language features and task performances with ProFiT for correlation analysis. Language features include typological & phylogenetic similarities (**Sim$_1$**), lexical similarities (**Sim$_2$**), and target language size (**Size**). Task performance contains the ProFiT results on the three datasets with both mBERT and XLM-R models.

mance, including (1) the size of the pretraining corpus for the target language and (2) the similarity between the source and target languages. We analyze how these two factors influence ProFiT 's effectiveness for the languages on three tasks.

The pretraining corpus size of the target languages can be simply measured by the log_2 of the number of articles in Wikipedia[2].

For measuring the similarity between languages, we employ methods from recent studies of language representations. In these studies, languages are encoded as vectors according to their various linguistic and typological features. With these language vectors, a range of distance metrics, such as Euclidean distance and cosine similarity, can be used to measure the similarity between languages. Littell et al. (2017) proposed LANG2VEC, which encodes languages using 5 vectors, with each vector representing a specific language feature. Östling and Kurfalı (2023) measured the lexical similarity by calculating language vectors based on the ASJP word list database (Wichmann et al., 2022). Liu et al. (2023d) recently proposed a novel language similarity metric from the perspective of conceptualization across multiple languages.

In our work, we compute two similarity metrics: (i) a comprehensive linguistic similarity metric based on LANG2VEC (Littell et al., 2017) and (ii) a lexical similarity metric based on the ASJP word list database (Östling and Kurfalı, 2023).

The LANG2VEC approach provides information-rich vector representations of languages from different linguistic and ethnological perspectives. We adopt five linguistic categories: syntax

[2] https://meta.wikimedia.org/wiki/List_of_Wikipedias

(SYN), phonology (PHO), phonological inventory (INV), language family (FAM), and geography (GEO). SYN, PHO, and INV are typological categories, and FAM and GEO are phylogenetic categories. Given these vectors, we calculate 5 different cosine similarity metrics between English and each target language.

The lexical similarity metric is based on a mean-normalized pairwise Levenshtein distance matrix from ASJP. The language vectors used for calculating the lexical similarity are reduced in dimensionality. Two dimensionality reduction methods are employed for calculating the lexical similarity: Uniform Manifold Approximation and Projection (*UMAP*) (McInnes et al., 2018) and Singular Value Decomposition (*SVD*) (Stewart, 1993).

The final typological and phylogenetic similarity score $\mathbf{Sim}_1$ for each language pair is calculated by averaging the 5 similarities of LANG2VEC. Similarly, the lexical similarity score $\mathbf{Sim}_2$ is calculated by averaging the similarities of the *UMAP* and *SVD* vectors. More formally, as Eq. 4.3 shows, let f denote a feature from the feature set $\mathcal{F}_n$ for metric n, and let v_f denote the corresponding feature vector. As introduced in §3.2.5.2, the sim_1 and sim_2 scores for the source language English (e) and some target language j are then calculated by:

$$sim_n(e,j) = \frac{1}{|\mathcal{F}_n|} \sum_{f \in \mathcal{F}_n} \frac{v_f(e) \cdot v_f(j)}{\|v_f(e)\|_2 \, \|v_f(j)\|_2} \tag{4.3}$$

Table 4.6 shows a list of language features (typological & phylogenetic similarities, lexical similarities, and target language size) and task performances with PROFIT for the following correlation analysis. The language similarities, namely the typological & phylogenetic similarities ($\mathbf{Sim}_1$) and lexical similarities ($\mathbf{Sim}_2$), refer to the similarity between each language and English, based on the above introduced language vectors. Sim_1 and Sim_2 are calculated by Eq. 4.3. *ja* and *zh* are not included in Östling and Kurfalı (2023)'s original language sets, thus these two values are missing for the lexical similarities. The target language size (**Size**) is calculated by the log_2 of the number of articles in Wikipedia.

Based on the obtained language features and experimental results of task performance with PROFIT , we did a correlation analysis. Table 4.7 shows the results of the two correlation tests on each task.

According to the results of Pearson and Spearman tests and the p values, the two factors, namely, both the size of pretraining data for the target language and the similarity of typological and phylogenetic features of languages (sim_1) have a significant positive correlation with the improvement of cross-lingual performance especially on XNLI, with both PROFIT -M and PROFIT -X models. Only the correlations calculated with the similarity of lexical features (sim_2) show some insignificant results. Furthermore, on XNLI, the correlation with language similarity is stronger with PROFIT -X, while the correlation with target data size is stronger with PROFIT -M. We argue that the XLM-R model is bigger than mBERT, so that the linguistic features have more effect on the performance, while for the smaller model mBERT, the data size plays a greater role, which further supports our findings in §4.1.4 that the applied pretrained model for fine-tuning has an impact on the PROFIT performance.

On PAWS-X and Amazon, we find weak correlations with the proposed factors, which could result from the limitation of languages in test data: XNLI comprises 15 different languages,

Task	Model	Stat.	sim$_1$		sim$_2$		Size	
			corr.	*p*	*corr.*	*p*	*corr.*	*p*
Amazon	ProFiT -M	P	0.73	0.16*	-0.95	0.21*	0.81	0.09*
		S	0.70	0.19*	-1.00	0.00	0.50	0.39*
	ProFiT -X	P	0.80	0.10*	1.00	0.01	0.92	0.03
		S	0.80	0.10*	1.00	0.00	1.00	1e-24
PAWS-X	ProFiT -M	P	0.82	0.05	0.31	0.69*	0.82	0.04
		S	0.83	0.04	0.20	0.80*	0.60	0.21*
	ProFiT -X	P	0.83	0.04	0.34	0.66*	0.84	0.04
		S	0.77	0.07*	0.20	0.80*	0.71	0.11*
XNLI	ProFiT -M	P	0.57	0.03	0.43	0.14*	0.86	9e-05
		S	0.59	0.03	0.53	0.06*	0.90	1e-05
	ProFiT -X	P	0.72	4e-03	0.43	0.14*	0.70	5e-03
		S	0.77	1e-03	0.63	0.02	0.72	4e-03

Table 4.7: Correlations between task performance and language similarities (sim$_1$ & sim$_2$) and target language size. P stands for Pearson's test and S for Spearman's test. Insignificant results with a p value > 0.05 are marked with *.

whereas PAWS-X and Amazon only contain 7 and 6 languages in the test set, respectively. Thus, weaker correlations have been found. To sum up, language similarity and size are two factors that impact the cross-lingual performance in our study, and we find significant correlations when the test set contains a larger number of languages.

4.1.6 Sum-Up

In our work, we introduce ProFiT for zero-shot cross-lingual transfer, a pipeline which reformulates input examples into cloze-style prompts and applies the input examples with the prompts and their verbalizers as masked tokens to fine-tuning, changing the sentence classification task of vanilla fine-tuning into a masked token prediction task. We fine-tune the multilingual pretrained language model (MPLM) on source language prompts and apply it to target language data. We use ProFiT with the two MPLMs mBERT and XML-R, and evaluate its efficacy on three different types of multilingual classification tasks in natural language understanding – multi-class sentiment classification, binary paraphrase identification, and multi-class natural language inference. Our experiments show that ProFiT outperforms vanilla fine-tuning with both mBERT and XML-R on all three tasks. We further discovered that the performance improvement of ProFiT is generally more obvious in few-shot scenarios. Additionally, we demonstrate that the similarity of the source and target language and the size of the target language pretraining data significantly correlate with the cross-lingual transfer performance of ProFiT , especially on a big dataset with a variety of test languages.

4.2 Token-Level Prompt Decomposition Fine-Tuning for Cross-Lingual Sequence Labeling Tasks

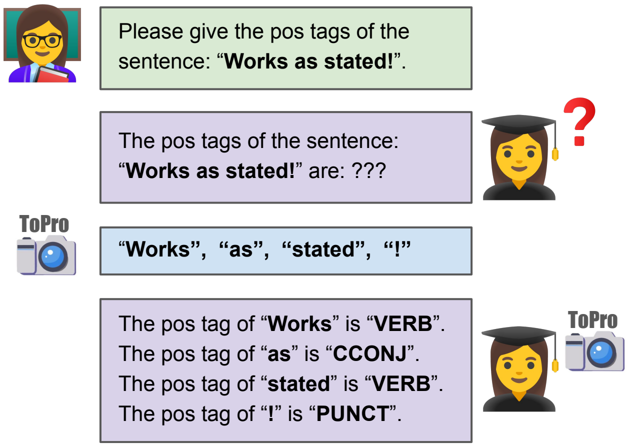
This section corresponds to the following work:

> Bolei Ma*, **Ercong Nie***, Shuzhou Yuan, Helmut Schmid, Michael Färber, Frauke Kreuter, and Hinrich Schuetze. 2024. ToPro: Token-Level Prompt Decomposition for Cross-Lingual Sequence Labeling Tasks. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2024. Volume 1: Long Papers), pages 2685–2702, St. Julian's, Malta. Association for Computational Linguistics.
> * equal contributions.

Declaration of Co-Authorship. I conceived the idea of applying prompt-based fine-tuning methods to cross-lingual sequence labeling tasks, and proposed the ToPro methodology of decomposing a prompt into a set of token-level single prompts. Bolei Ma implemented ToPro and validated it with experiments on UDPOS and WikiANN datasets using two encoder models. I ran the baseline experiments of prompt-tuning. Shuzhou Yuan implemented ToPro with the encoder-decoder mT5 model. Bolei Ma and I drafted the paper together. The other authors on the list are supervisors, who supervise the project process and provide valuable feedback.

Summary of This Section

Prompt-based methods have been successfully applied to multilingual pretrained language models for zero-shot cross-lingual understanding (§4.1). However, most previous studies primarily focused on sentence-level classification tasks, and only a few considered token-level labeling tasks such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging. In this section, we introduce **To**ken-Level **Pro**mpt Decomposition (**ToPro**), which facilitates the prompt-based method for token-level sequence labeling tasks. The ToPro method decomposes an input sentence into single tokens and applies one prompt template to each token. Our experiments on multilingual NER and POS tagging datasets demonstrate that ToPro -based fine-tuning outperforms Vanilla fine-tuning and Prompt-Tuning in zero-shot cross-lingual transfer, especially for languages that are typologically different from the source language, English. Our method also attains state-of-the-art performance when employed with the mT5 model. Besides, our exploratory study in multilingual large language models shows that ToPro performs much better than the current in-context learning method. Overall, the performance improvements show that ToPro could potentially serve as a novel and simple benchmarking method for sequence labeling tasks.



Please give the pos tags of the sentence: "**Works as stated!**".

The pos tags of the sentence: "**Works as stated!**" are: ???

"**Works**", "**as**", "**stated**", "**!**"

The pos tag of "**Works**" is "**VERB**".
The pos tag of "**as**" is "**CCONJ**".
The pos tag of "**stated**" is "**VERB**".
The pos tag of "**!**" is "**PUNCT**".

Figure 4.4: ToPro as a token-level prompting method for sequence labeling tasks. It decomposes the input sentence into single tokens and applies the prompt template to each token, inspired by human step-by-step logical thinking when solving this kind of task.

4.2.1 Motivation and Introduction

As multilingual pretrained language models (MPLMs) continue to evolve (Devlin et al., 2019; Conneau et al., 2020; Liu et al., 2020; Xue et al., 2021; Shliazhko et al., 2022), zero-shot cross-lingual transfer methods are gaining increasing popularity within the multilingual NLP domain (Lauscher et al., 2020; Nie et al., 2023a). In light of the limited availability of training data in many low-resource languages, prior research (Artetxe et al., 2020; Hu et al., 2020b) employed zero-shot cross-lingual transfer learning by fine-tuning an MPLM on a high-resource language such as English, and then directly applying the fine-tuned system to low-resource languages.

Prompt-based learning (Schick and Schütze, 2021a,b,c) is steadily garnering traction in recent NLP research. Prompt-based methods reformulate downstream tasks as language modeling tasks by using prompts comprising a template and a set of label words. The prompt can be either discrete in a textual format or continuous, performing prompting directly in the embedding space of the model (Liu et al., 2023a). Much recent work highlights that applying prompt-based fine-tuning to MPLMs enables better zero-shot cross-lingual transfer performance (Zhao and Schütze, 2021; Huang et al., 2022; Nie et al., 2023b; Zhou et al., 2023). However, they focus on sentence-level classification tasks such as sentiment analysis (Keung et al., 2020), XNLI (Conneau et al., 2018), and paraphrase detection (Zhang et al., 2019a). Token-level sequence labeling tasks like Named Entity Recognition (NER) and Part-of-Speech (POS) Tagging rarely benefit from the advantages of prompt-based fine-tuning, primarily due to the intricate challenge of devising an appropriate prompt template.

To enhance the applicability of prompt-based learning to token-level sequence labeling tasks, we introduce the **To**ken-Level **Pro**mpt Decomposition (**TOPRO**) method. TOPRO splits the input sentence into tokens and creates a separate prompt for each token, which asks for its label, following the human step-by-step logical thinking when solving these tasks, as shown in Figure 4.4. The evaluation on NER and POS tagging tasks shows that the TOPRO -based fine-tuning achieves stronger zero-shot cross-lingual transfer performance than Vanilla fine-tuning and Prompt-Tuning, especially for languages that are typologically different from the source language (English).

To sum up, our contributions are as follows:

1. We propose a novel and simple method, called TOPRO , which improves zero-shot cross-lingual transfer in token-level sequence labeling tasks by taking advantage of prompt-based learning.

2. We substantiate the strength of TOPRO in zero-shot cross-lingual fine-tuning through evaluations on NER and POS tagging tasks. Our method not only outperforms baselines for over 40 languages but also demonstrates efficacy in zero-shot English ICL, making it a promising benchmarking method for MLLMs in token-level tasks.

3. We conduct a thorough cross-lingual analysis, revealing that TOPRO exhibits particularly strong performance for languages that are typologically different from the source language, English.

4.2.2 TOPRO for Fine-Tuning

Problem Formulation In prompt-based learning, there is a pattern-verbalizer pair (PVP) (Schick and Schütze, 2021a) consisting of (i) a *prompt pattern* which converts the input text into a cloze-style question with a mask token, and (ii) a *verbalizer* which maps the labels onto representative words from the LM's vocabulary. This aligns well with the nature of text classification tasks, where one label is predicted based on the input text. As Figure 4.5 shows, the input text

X of a sentiment analysis task can be reformulated with a prompt pattern $P(\cdot)$ into a prompted input representation $P(X)$ = " Works as stated! In summary, the product was [MASK]. " The

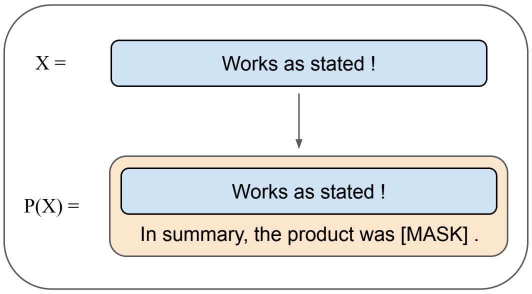


Figure 4.5: A prompt example for text classification.

prompt $P(X)$ is processed by the LM to determine the most likely verbalizer word in the masked position. The label corresponding to this verbalizer is the prediction, which is evaluated against the gold standard.

However, in sequence labeling tasks, each token of the input should receive a label. Thus, it is not possible to apply this type of prompt pattern with one mask token directly for token classification.

Token-Level Prompt Decomposition (ToPro) When given such a token-level sequence labeling task, a human usually solves the task token by token. Inspired by this human process as well as the prompt design for sentence classification tasks, we propose a new prompting method ToPro for token classification which decomposes an input sentence into tokens and generates a series of prompts – one prompt for each token. Let $X = x_1, x_2, ..., x_m$ denote an input sentence consisting of m tokens. Our prompt generator function $P(T, X)$ generates m prompts by filling the template $T(\cdot, \cdot)$ with the sentence X and each of the tokens $x_1, x_2, ..., x_m$, respectively.

$$P(T, X) = \{T(X, x_1), ..., T(X, x_m)\} \tag{4.4}$$

Figure 4.6 shows the prompts generated by $P(T, X)$ for the input X = "Works as stated !" and the template $T(X, x_i)$ = " X The POS tag of x_i is a kind of [MASK] ."

Prompt-Based Fine-Tuning and Cross-Lingual Transfer Following the previous section §4.1, we conduct prompt-based fine-tuning to evaluate our ToPro approach in a zero-shot cross-lingual context. Let $D = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ denote the set of training examples in the source language, where $X_1, \ldots, X_n$ are token sequences and $Y_1, \ldots, Y_n$ are tag sequences. Given $(X, Y) \in D$, the ToPro function $P(T, X)$ reformulates the input sentence X into a set of cloze-style questions $\{T(X, x_1), ..., T(X, x_m)\}$ with masked tokens. The pretrained language model M with trainable parameters θ performs masked token prediction and returns the probabilities $p(\cdot) = M(T(X, x_i), \theta)$ of all candidate words for the masked token in the prompt $T(X, x_i)$. The verbalizer function $V(\cdot)$ is a bijective mapping from the set of class labels L to a set of

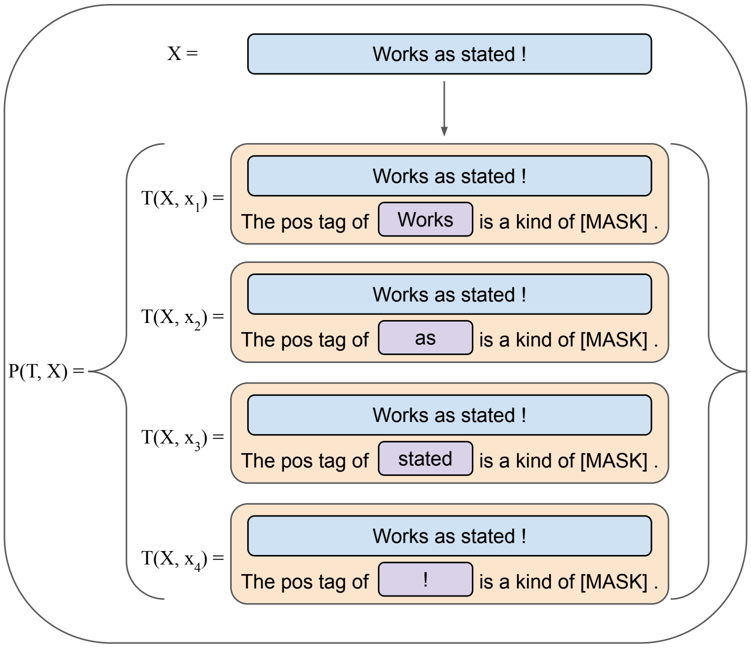

X =
Works as stated !
P(T, X) =
T(X, x1) =
Works as stated !
The pos tag of Works is a kind of [MASK] .
T(X, x2) =
Works as stated !
The pos tag of as is a kind of [MASK] .
T(X, x3) =
Works as stated !
The pos tag of stated is a kind of [MASK] .
T(X, x4) =
Works as stated !
The pos tag of ! is a kind of [MASK] .

Figure 4.6: An example of ToPro framework for sequence labeling.

verbalizers from the source language vocabulary. For each token, we predict the tag $\hat{y}$ whose verbalizer $V(\hat{y})$ receives the highest probability from model M:

$$\hat{y} = \arg\max_{y \in L} p(V(y)) \tag{4.5}$$

We fine-tune the parameters θ of model M by minimizing the cross-entropy loss function $\ell(D, \theta)$:

$$\ell(D, \theta) = -\sum_{(X,Y)\in D} \sum_{i=1}^{|Y|} \log M(T(X, x_i), \theta)(V(y_i)) \tag{4.6}$$

The fine-tuned model is used to predict the labels of the target language examples $\{X'_1, ..., X'_n\}$ using the same prompt pattern $T(\cdot, \cdot)$ and verbalizer $V(\cdot)$ as during fine-tuning . The best tags Y'_j for each example X'_j are predicted according to Eq. 4.5.

4.2.3 Experimental Setups

4.2.3.1 Datasets and Prompt Designs

We chose the following two representative datasets for sequence labeling tasks:

PAN-X, also called WikiANN, is a multilingual NER dataset based on Wikipedia articles, including 282 languages (Pan et al., 2017). In our work, we use the subset of 48 languages that is part of the XTREME benchmark (Hu et al., 2020b) to facilitate comparisons with related work.

For each token x_i of an input sequence X, we use the following prompt template $T(X, x_i)$:

> $T(X, x_i) = X \circ$ “ The named entity of ” $\circ\ x_i \circ$ “ is a kind of: [MASK].”

The PAN-X dataset is annotated with location (LOC), person (PER), and organization (ORG) in IOB2 format. These labels are difficult to understand for the language model. Therefore, we replace them with real words and train the model to predict those instead. However, the model can only predict single words from its vocabulary as fillers for the [MASK] position. So, we choose the replacement words from the model's vocabulary.

As IOB2 annotates the beginning of a name and its remaining tokens with different tags, we use a word and its hyponym to represent the beginning of a name and its remaining tokens, respectively. For instance, we use the hypernym “location” for the beginning of the LOC and the hyponym “place” for the other words which should be semantically inside of the term “location”. The verbalizer function $V(\cdot)$ for tag set Y is defined as follows:

V(B-LOC) = location	V(I-LOC) = place
V(B-ORG) = organization	V(I-ORG) = body
V(B-PER) = person	V(I-PER) = name
V(O) = other	

UDPOS was extracted from the Universal Dependency treebanks (Zeman et al., 2019). It contains 38 languages and is part of the XTREME benchmark (Hu et al., 2020b).

Similarly to the PAN-X dataset, we use the following prompt template $T(X, x_i)$ for token x_i of an input sequence X by paraphrasing the tags with semantically related words:

> $T(X, x_i) = X \circ$ “ The pos tag of ” $\circ\ x_i \circ$ “ is a kind of: [MASK].”

We define the verbalizer $V(\cdot)$ for the 14 tags as follows:

V(ADJ) = modification	V(ADP) = position
V(ADV) = verbal	V(AUX) = auxiliar
V(CCONJ) = link	V(DET) = determine
V(INTJ) = mode	V(NOUN) = thing
V(NUM) = number	V(PART) = functional
V(PRON) = reference	V(PROPN) = name
V(PUNCT) = punct	V(SCONJ) = condition
V(SYM) = symbol	V(VERB) = verb
V(X) = other	

The tags of the two datasets and their detailed meanings are documented in Table 4.8 and Table 4.9. We cannot select words like “adjective” and “adverb” which would better represent the meanings of the tags, because the verbalizers have to come from the vocabulary of the PLM so that the masked language model is able to predict them as a single unit. Instead, we use semantically related words from the vocabulary as verbalizers.

Tags	Meaning
B-LOC	location (beginning)
B-ORG	organization (beginning)
B-PER	person (beginning)
I-LOC	location (inside)
I-ORG	organization (inside)
I-PER	person (inside)
O	other

Table 4.8: IOB2 tags

Tags	Meaning
ADJ	adjective
ADP	adposition
ADV	adverb
AUX	auxiliary
CCONJ	coordinating conjunction
DET	determiner
INTJ	interjection
NOUN	noun
NUM	numeral
PART	particle
PRON	pronoun
PROPN	proper noun
PUNCT	punctuation
SCONJ	subordinating conjunction
SYM	symbol
VERB	verb
X	other

Table 4.9: Universal POS tags

4.2.3.2 Baselines

We compare our approach with the following baselines:

Vanilla Fine-Tuning (Vanilla) The vanilla fine-tuning method predicts the token labels through the hidden embeddings of each token in the output layer without using a prompt pattern. We use the cross-entropy loss as the objective function for fine-tuning and `AdamW` for optimization with a learning rate of 1e-5. The fine-tuned models are used to predict the test data.

Prompt-Tuning (PT) Prompt-Tuning only trains a small number of parameters, e.g., a continuous prompt or a task classifier Lester et al. (2021); Liu et al. (2022c). We implement the prompt-tuning method of Tu et al. (2022) for zero-shot cross-lingual transfer by tuning the prefix prompts and layer prompts for the two sequence labeling tasks.

4.2.3.3 Multilingual Models

The following MPLMs from the HuggingFace Transformers library (Wolf et al., 2020) are applied in our main experiments:

Encoder-Only Models For encoder-only models, we use the multilingual BERT model (Devlin et al., 2019) `bert-base-multilingual-cased` (B) and the XLM-R model (Conneau et al., 2020) `xlm-roberta-base` (X).

Encoder-Decoder Model We use multilingual T5 model (Xue et al., 2021) `mt5-base` (T) as the encoder-decoder model representative. We include mT5 in our experiments, as we wish to explore the potential of ToPro with different types of models. To align with the text-to-text transformer format, we have redefined the output structure for both NER and POS tasks, drawing inspiration from the prior work of mT5 Xue et al. (2021). For the input text, we introduce task descriptions as prompts, specifically "NER tagging:" for the PAN-X dataset and "POS tagging:" for the UDPOS dataset. Regarding the target text, we append tags to each token and insert delimiters between tokens to create a coherent sequence of text. The following example illustrates our preprocessing procedure using a sample from the UDPOS dataset for Vanilla fine-tuning:

- **Input text:** POS tagging: On the other hand, it looks pretty cool .
- **Target text:** ADP: On $$ DET: the $$ ADJ: other $$ NOUN: hand $$ PUNT: , $$ PRON: it $$ VERB: looks $$ ADV: pretty $$ ADJ: cool $$ PUNT: .

As for the ToPro method, we use the same prompt pattern as for encoder-only models:

- **Input text:** On the other hand, it looks pretty cool . The pos tag of On is:
- **Target text:** ADP

4.2.4 Results and Analysis

Main Results Table 4.10 gives an overview of the average results[3] on PAN-X and UDPOS. We find that ToPro Fine-Tuning outperforms Vanilla Fine-Tuning and Prompt-Tuning obviously on

[3] Since we are interested in the zero-shot cross-lingual transfer performance, we do not include the English results in the average performance. Our evaluation metric is the weighted average F1-score.

both tasks in mBERT and XLM-R settings: On PAN-X, the performance is improved by **19.18%** and **25.16%** compared to Vanilla and Prompt-Tuning respectively, when trained with mBERT, and by **18.73%** and **26.98%** with XLM-R. On UDPOS, the performance is improved by **5.27%** and **6.24%** compared to Vanilla and Prompt-Tuning, respectively, when trained with mBERT, and by **3.74%** and **4.3%** with XLM-R.

In the mT5 setting, the TOPRO Fine-Tuning outperforms Vanilla Fine-Tuning on both tasks as well, namely by **28.63%** on PAN-X, and by **14.72%** on UDPOS. We notice that the mT5 model performs even better than the two encoder-only models and achieves SOTA performance[4], showing the potential of TOPRO with different model types. We find that Prompt-Tuning does not work well with mT5, as it requires more training epochs for the model to achieve subtle performance improvements, necessitating even longer training time compared to the Vanilla baselines. One possible reason for this could be the limited number of trainable parameters in mT5 with Prompt-Tuning, as only 0.002% of the parameters are updated with our current prompt settings. We exclude the results of Prompt-Tuning for mT5 because the increased training resources do not align with the efficiency-focused goals of Prompt-Tuning as a training methodology.

When comparing performances on the two tasks generally, we notice that the performance shows greater improvement on PAN-X with all three models, indicating that the NER task PAN-X has a greater improvement potential.

Model	Method	PAN-X	UDPOS
mBERT	Vanilla Fine-Tuning	62.73	70.89
	Prompt-Tuning	56.76	69.91
	TOPRO Fine-Tuning	**81.91**	**76.16**
XLM-R	Vanilla Fine-Tuning	61.30	72.42
	Prompt-Tuning	53.05	71.86
	TOPRO Fine-Tuning	**80.03**	**76.16**
mT5	Vanilla Fine-Tuning	64.19	71.38
	Prompt-Tuning	-*	-*
	TOPRO Fine-Tuning	**92.82**	**86.11**

Table 4.10: Overview of average results on PAN-X and UDPOS. $*$: The results of PT with mT5 are excluded from the comparison as the F1 scores are 0 for the current parameter settings.

4.2.5 Cross-Lingual Transfer Analysis

The detailed results of the cross-lingual transfer performance of TOPRO compared to the baselines for each target language are documented in Appendix E. Table 4.11 and Table 4.12 show

[4] Based on the evaluation results available at `https://sites.research.google/xtreme/dataset`, as of Jan. 23, 2024, the SOTA performance in structured prediction, calculated as the mean value of PANX-X and UDPOS, is 84.6. Our mT5 model, when used with TOPRO , achieves an impressive score of 89.47.

langs	B (Vanilla)	B (PT)	X (Vanilla)	X (PT)	T (Vanilla)
en	8.96	13.71	10.90	16.27	19.38
af	12.81	19.50	15.00	20.10	19.82
ar	18.52	23.10	20.57	24.09	39.14
az	17.73	21.83	22.65	25.46	32.78
bg	11.29	16.33	11.17	16.05	23.13
bn	8.24	19.52	3.10	18.65	30.42
de	13.30	18.26	17.15	23.13	21.01
el	18.03	26.54	16.28	27.09	19.34
es	10.99	16.88	13.13	18.42	26.08
et	12.11	16.23	17.53	22.83	21.55
eu	19.91	24.36	26.52	36.62	27.50
fa	27.10	34.67	14.09	24.17	47.48
fi	12.51	17.24	15.29	20.42	20.78
fr	6.75	12.13	10.39	17.06	20.87
gu	33.33	55.16	30.99	40.57	31.99
he	27.47	31.26	30.94	38.85	24.09
hi	12.70	18.50	11.18	18.71	30.79
hu	14.76	20.04	14.96	21.21	22.97
id	16.79	19.60	21.31	24.03	26.95
it	10.15	13.13	11.78	17.81	19.03
ja	41.04	45.53	47.61	49.89	43.52
jv	18.70	23.05	16.43	32.80	22.80
ka	19.31	25.80	20.48	30.28	25.85
kk	33.74	34.89	42.36	42.48	28.63
ko	22.33	25.43	31.42	37.05	33.16
lt	13.58	18.13	14.24	21.01	23.48
ml	26.57	32.21	25.70	34.47	32.55
mr	25.15	31.75	21.01	33.32	31.70
ms	14.50	18.38	8.25	28.53	17.64
my	29.29	39.47	31.69	40.16	48.96
nl	10.12	14.57	12.32	17.12	19.50
pa	25.38	28.31	19.42	35.89	32.47
pl	10.12	13.49	13.02	17.62	21.03
pt	7.48	13.25	9.15	15.87	23.98
qu	12.97	31.44	17.08	32.22	25.16
ro	7.91	22.31	13.15	24.11	25.06
ru	19.37	26.56	18.10	25.80	27.92
sw	9.25	18.76	7.81	19.75	25.30
ta	24.48	28.73	26.68	33.46	29.84
te	32.97	36.06	36.53	43.85	28.14
th	67.60	67.84	16.48	15.89	50.10
tl	11.40	11.00	8.52	16.21	27.06
tr	12.63	20.13	13.77	24.87	26.93
uk	14.63	20.74	12.30	24.53	23.51
ur	29.96	36.69	1.63	22.93	51.31
vi	16.35	18.87	14.26	20.49	31.65
yo	15.41	27.00	16.13	30.82	23.30
zh	24.88	27.66	40.81	41.58	39.50
avg.	19.18	25.16	18.73	26.98	28.63

Table 4.11: Performance difference (−) of TOPRO to Vanilla or Prompt Tuning (PT) with mBERT (B), XLM-R (X), and mT5 (T) on PAN-X.

the performance improvements of TOPRO compared to the baselines for each language. Overall, TOPRO -based Fine-Tuning outperforms Vanilla Fine-Tuning and Prompt-Tuning on average. However, we can notice individual performance differences between the languages.

On both tasks, we find that the performance gain of TOPRO for English (en) is among the lowest across all languages. Since English is the language on which the models have been fine-tuned, we conclude that TOPRO is particularly effective in cross-lingual zero-shot scenarios. The reason could be that the models are only fine-tuned on the English dataset. Therefore, TOPRO 's potential performance improvement is smaller for English than for other languages.

langs	B (Vanilla)	B (PT)	X (Vanilla)	X (PT)	T (Vanilla)
en	0.54	0.87	0.41	0.87	7.90
af	3.27	3.31	2.00	1.96	7.16
ar	16.51	14.43	4.65	4.37	15.23
bg	2.80	2.63	0.56	0.69	14.34
de	3.11	3.44	1.58	1.85	12.48
el	3.80	4.86	-0.49	-0.64	13.06
es	-0.86	0.89	-1.23	-0.89	5.73
et	3.86	7.90	0.94	1.94	11.58
eu	10.11	8.87	1.88	5.24	14.14
fa	2.23	1.42	0.82	1.47	15.12
fi	2.63	5.21	0.48	1.21	11.68
fr	-0.16	4.76	-5.36	-5.00	8.41
he	24.40	24.55	14.12	14.38	23.68
hi	9.61	8.62	3.63	3.80	18.68
hu	0.56	1.08	-2.07	-1.82	13.90
id	4.63	4.83	4.10	4.43	13.81
it	-2.01	-0.33	-1.25	-2.35	8.28
ja	5.06	5.34	29.16	32.61	27.31
kk	4.45	4.79	0.46	1.67	15.61
ko	13.85	13.04	11.39	10.85	25.57
lt	3.75	6.61	2.49	4.05	12.81
mr	5.39	8.51	-2.52	-1.13	17.19
nl	0.50	1.01	0.29	0.60	9.15
pl	3.20	3.82	1.84	1.50	13.76
pt	-1.07	-0.66	-0.79	-0.75	7.96
ro	3.15	4.01	1.46	1.89	14.07
ru	4.20	3.44	1.54	2.19	11.36
ta	13.43	12.89	10.85	10.88	20.38
te	-0.93	1.45	-0.59	1.68	12.00
th	15.83	19.92	25.28	29.21	15.61
tl	-0.59	4.12	-4.68	-6.53	19.38
tr	1.82	5.11	-0.37	1.11	16.88
uk	5.91	5.62	1.95	2.32	13.51
ur	12.04	11.07	7.94	6.14	20.46
vi	2.79	4.08	1.30	2.44	20.61
wo	2.45	4.19	-10.70	-9.42	-0.88
yo	5.74	7.79	-6.59	-5.54	5.97
zh	9.56	8.29	44.30	42.58	38.53
avg.	5.27	6.24	3.74	4.30	14.72

Table 4.12: Performance difference (−) of TOPRO to Vanilla or PT with mBERT (B), XLM-R (X), and mT5 (T) on UDPOS.

Another explanation could be that the accuracy on English is the highest, therefore, the potential for improvement is lower: Raising an accuracy of 90% by 10% is much harder than raising an accuracy of 50% by 10%.

On PAN-X , TOPRO outperforms Vanilla and Prompt-Tuning across all target languages, with some language-independent variations. The improvements in languages such as Persian (fa), Gujarati (gu), Hebrew (he), Japanese (ja), Kazakh (kk), Burmese (my), Telugu (te), Thai (th), Urdu (ur), and Chinese (zh) are above the average. All these languages are from different language groups than English and have different writing systems. We can conclude that the performance improvement of TOPRO is particularly high for languages that differ a lot from English, further indicating the cross-lingual ability of our prompt-based method.

On UDPOS , TOPRO outperforms Vanilla and PT in most of the languages, although there are some languages for which TOPRO performs slightly worse and the overall performance gain is not as high as on PAN-X. Typically, the improvements for languages such as Arabic (ar), Basque (eu), Hebrew (he), Korean (ko), Tamil (ta), Thai (th), Urdu (ur), and Chinese (zh) are above average. The improvements over Vanilla in Chinese reach 44.3% and 38.53% for XLM-R and mT5, respectively, and the improvement over PT in Chinese is 42.58%.

Overall, the results show that TOPRO outperforms Vanilla and PT on both sequence labeling tasks, indicating that the TOPRO method has a better ability to transfer knowledge cross-lingually. And the NER performance is even better than the performance for POS tagging. When analyzing the performances for individual languages, we find that TOPRO has a strong performance for zero-shot cross-lingual transfer, particularly in languages with low similarity to English and different writing systems. The prompt-based approach seems to mitigate the language barriers and facilitate cross-lingual transfer. Additionally, the results vary across target languages, highlighting the importance of language typology and writing systems in determining the effectiveness of TOPRO .

4.2.6 Error Analysis

In this section, we analyze selected instances from the UDPOS task with typical annotation errors by the models in Table 4.13.

The first example is a sentence in Chinese (zh), which is typologically and orthographically quite different from the training language, English. The first two tokens marked red in this example 是 ("be") 指 ("refer to") are a pair of verbs, a so-called double-verb structure. They are both predicted by Vanilla as PUNCT (punctuation), but by TOPRO as AUX (auxiliary) and VERB, which are quite close to the correct tags, as the auxiliary itself is a special kind of verb. The tokens 長 ("long") 約 ("around") are predicted by Vanilla again as PUNCT, but correctly by TOPRO as ADJ and ADV. Moreover, the tokens 海岸 ("coast") 線 ("line") are predicted by Vanilla still as PUNCT, and by TOPRO as PROPN (proper noun) and NOUN, which, though not the same as the original tags NOUN and PART (particle), are already close to the original tags as they are all a kind of noun. In this case, we notice that the Vanilla model tends to predict PUNCT for the majority of the tokens, whereas the TOPRO method often predicts the correct tags or at least semantically related tags.

The second example is in Japanese (ja), which is also typologically and orthographically quite different from English. The first token 政府 ("government") is predicted by Vanilla as DET (determiner) which is somehow close to its original tag, and correctly by TOPRO as NOUN. The token もっと ("more") is predicted by Vanilla as PUNCT, but correctly by TOPRO as ADV. The token し ("that") is originally AUX, and it is predicted by Vanilla again as PUNCT, but by TOPRO as VERB, which is already close to the meaning of AUX. And the token pair 排除 ("exclude") す ("do") has original labels VERB AUX, and is predicted by TOPRO as VERB VERB, which are still very close to their original labels. However, Vanilla predicts them again as PUNCT PUNCT. Similar to the Chinese example, the Vanilla method tends to predict PUNCT for unfamiliar tokens, whereas TOPRO generates the correct tags or at least tags close to the correct tags.

Input sequence & Its Gloss & Tags (True, Vanilla, TOPRO) & Translation
Case 1 (zh) **Input:** 温暖 海岸 是 指 西班牙 穆爾西亞 自治 區 長 約 250 公里 的 地中 海 海岸 線 。 **Gloss:** Warm coast be refer Spain Murcia autonomous region long approximately 250 km 's Mediterranean sea coast line . **True:** propn noun verb verb propn propn verb part adj adv num noun part propn part noun part punct **Vanilla:** propn punct punct punct punct propn punct punct punct punct num num punct noun noun punct punct punct (0.19 F1) **TOPRO :** propn propn aux verb propn propn propn propn adj adv num noun adp noun noun propn noun punct (0.47 F1) **Translation:** The Costa Cálida is the 250-kilometer-long Mediterranean coastline of the Autonomous Region of Murcia, Spain.
Case 2 (ja) **Input:** 政府 が もっと 宗教法人 の 定義 を 厳しく し , こういう 団体 は 排除 す べき 。 **Gloss:** government subject more religious organisation of definition object strictly that , such association topic exclude do must . **True:** noun adp adv noun adp noun adp adj aux punct adj noun adp verb aux aux punct **Vanilla:** det punct punct punct punct noun punct punct punct punct punct punct punct verb punct punct punct (0.29 F1) **TOPRO :** noun part adv noun part noun pron adj verb punct adj noun pron verb verb adj punct (0.64 F1) **Translation:** "The government should tighten the definition of religious corporations and eliminate such organizations."
Case 3 (de) **Input:** „ Mich interessiert etwas , wenn es mich zu Teilnahme zu erregen weiß ", sagte er in einem deutschen Fernsehen . **Gloss:** " me interest something , if it me to attendance to irritate knows " , said he in a German television . **True:** punct pron verb pron punct sconj pron pron adp noun part verb verb punct punct verb pron adp det adj noun punct **Vanilla:** punct pron verb pron punct sconj pron pron adp noun part verb verb punct punct verb pron adp det adj noun punct (1.00 F1) **TOPRO :** punct pron verb pron punct sconj pron pron part noun part verb verb punct punct verb pron adp det adj noun punct (0.95 F1) **Translation:** "I am interested in something if it knows how to excite me to participate", he said on German television."
Case 4 (nl) **Input:** „ We hebben een concept nodig voor verandering ", zei Djindjic in een interview met de Duitse televisie . **Gloss:** " we have a concept necessary for change " , said Djindjic in a interview with the German television . **True:** punct pron verb det noun adj adp noun punct punct verb propn adp det noun adp det adj noun punct **Vanilla:** punct pron verb det noun verb adp noun punct punct verb propn adp det noun adp det adj noun punct (0.95 F1) **TOPRO :** punct pron verb det noun verb adp noun punct punct verb propn adp det noun adp det adj noun punct (0.95 F1) **Translation:** "We need a concept for change", Djindjic said in an interview with German television.

Table 4.13: Comparison of the output of TOPRO and Vanilla for selected UDPOS examples with XLM-R. The interesting tokens and their tags are marked red. The sentences were translated into English using www.deepl.com.

The third example is an input sentence in German, which is very close to the English language. We reach, therefore, very high F1 scores both with Vanilla and TOPRO approaches. Noticeably, in this example, there is one token zu ("to") with two different kinds of POS: ADP and PART. The Vanilla correctly detects the difference, while TOPRO classifies both tokens as PART. This is a shortcoming of TOPRO's token-wise prompting strategy, which generates identical prompts for both occurrences of "zu".

The fourth example is a Dutch (nl) input sentence. Dutch is also closely related to English. We reach a high F1 score of 0.95 with both Vanilla and TOPRO approaches. The two approaches make the same error by predicting the token nodig ("necessary") as VERB, which should be ADJ.

In conclusion, the first two examples show that TOPRO works much better than Vanilla for languages that are typologically different from the source language of training. And even when predicting false POS tags, TOPRO tends to predict tags semantically close to the correct tags. The last two examples show the slightly worse performance of TOPRO for languages that are close to the source language of training. These findings support our claim in §4.2.5.

4.2.7 Sum-Up

In our work, we introduce TOPRO for token-level sequence labeling tasks, a novel and simple method that adopts the basic framework of prompting from sentence classification tasks and applies the prompt template to each token in a sentence. We evaluate the TOPRO -based fine-tuning for zero-shot cross-lingual transfer and compare it to Vanilla fine-tuning and Prompt-Tuning baselines. We apply TOPRO with three MPLMs on two representative sequence labeling tasks: NER and POS tagging. Our experiments show that TOPRO outperforms the baselines with the MPLMs and achieves SOTA performance with mT5. We further discovered that the performance improvement of TOPRO is generally more obvious in the cross-lingual context, especially for languages that are linguistically very different from the source language, English, highlighting its cross-lingual ability. Additionally, we applied the TOPRO method to MLLMs and noticed better performances of TOPRO as well, compared to existing benchmarking work. Overall, TOPRO shows a noticeable performance improvement and could serve as a potential benchmark for sequence labeling tasks for future studies in prompt-based learning.

4.3 Zero-Shot Transfer for Constituency Parsing of Historical German

This section corresponds to the following work:

> **Ercong Nie**, Helmut Schmid, and Hinrich Schütze. 2023. Cross-Lingual Constituency Parsing for Middle High German: A Delexicalized Approach. In Proceedings of the Ancient Language Processing Workshop (ALP 2023), pages 68–79, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Declaration of Co-Authorship. Helmut Schmid and I proposed the idea of applying zero-shot cross-lingual transfer with delexicalization for the constituency parsing of historical German. I developed the methodology pipeline, preprocessed the data, trained the delexicalization model on Modern German treebanks, evaluated it on the test set of Middle High German parse trees, and conducted the result analysis and case study. Helmut Schmid trained the POS taggers for Modern German and Middle High German. Helmut Schmid and Hinrich Schütze are supervisors and provided valuable feedback.

Summary of This Section

The technique of zero-shot cross-lingual transfer not only benefits modern languages, as indicated by the previous two sections (§4.1 and §4.2), but also facilitates historical language research. In this section, we apply zero-shot cross-lingual transfer to constituency parsing tasks and train a delexicalized constituency parser for a typical historical German language, i.e., Middle High German. Constituency parsing plays a fundamental role in advancing natural language processing (NLP) tasks. However, training an automatic syntactic analysis system for ancient languages solely relying on annotated parse data is a formidable task due to the inherent challenges in building treebanks for such languages. It demands extensive linguistic expertise, leading to a scarcity of available resources. To overcome this hurdle, cross-lingual transfer techniques that require minimal or even no annotated data for low-resource target languages offer a promising solution. In this study, we focus on building a constituency parser for **M**iddle **H**igh **G**erman (**MHG**) under realistic conditions, where no annotated MHG treebank is available for training. In our approach, we leverage the linguistic continuity and structural similarity between MHG and **M**odern **G**erman (**MG**), along with the abundance of MG treebank resources. Specifically, by employing the *delexicalization* method, we train a constituency parser on MG parse datasets and perform cross-lingual transfer to MHG parsing. Our delexicalized constituency parser demonstrates remarkable performance on the MHG test set, achieving an F1-score of 67.3%. It outperforms the best zero-shot cross-lingual[5] baseline by a margin of 28.6% points. These encouraging results underscore the practicality and potential for automatic syntactic analysis in other ancient languages that face similar challenges to MHG.

4.3.1 Background

Constituency parsing, which involves analyzing the grammatical structure of sentences and identifying the hierarchical relationships between words, plays a crucial role in linguistic research, especially for the analysis of ancient languages that are no longer spoken. Its significance extends beyond linguistic analysis, serving as a building block for various natural language processing (NLP) applications, such as information extraction (Jiang, 2012; Jiang and Diesner, 2019), sentiment analysis (Li et al., 2020), question answering (Hermjakob, 2001), etc. However, ancient languages lack large labeled and unlabeled corpora (Assael et al., 2022) and treebanks suitable for parser training are seldom available. This scarcity of resources can be attributed to two reasons. Firstly, ancient languages usually have a dearth of digital text resources. Secondly, the construction of a treebank for an ancient language requires substantial linguistic expertise and manual effort. Nonetheless, the continuity in the process of language evolution gives rise to linguistic similarities between ancient languages and their corresponding modern counterparts (Parravicini and Pievani, 2018). Cross-lingual transfer techniques (Ruder, 2019; Lauscher et al., 2020) are trained on high-resource languages and require little or no annotated data from

[5] As is prevalent in the realm of multilingual NLP, the term "zero-shot cross-lingual" in this context pertains to a transfer learning method where we finetune the model with task-specific data in a source language and test on the target language directly (Sitaram et al., 2023).

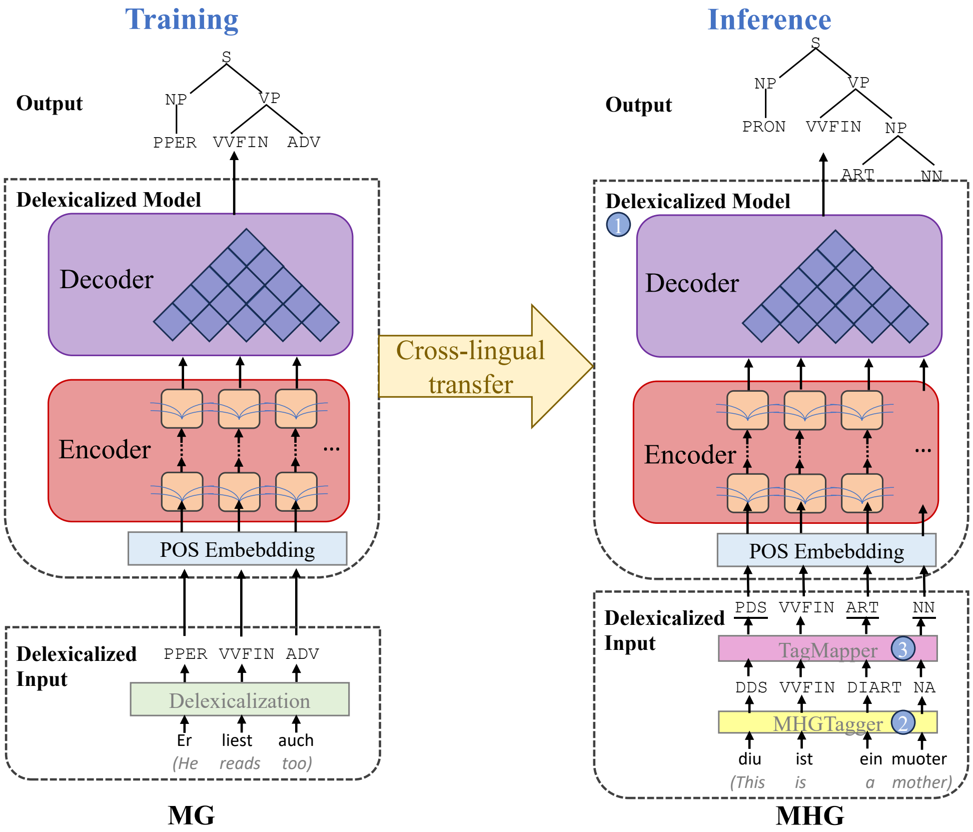


Figure 4.7: Overview of the cross-lingual delexicalized parsing system for MHG. In the training, the delexicalized parsing model is trained on the delexicalized MG trees. The trained parser is subsequently applied to MHG sentences. The delexicalized parsing system for MHG consists of three key modules: (1) *Delexicalized parsing model* trained on delexicalized MG trees, (2) *MHG POS tagger*, and (3) *Tag mapper*.

low-resource target languages. They can effectively be applied to languages with similar sentence structure and word order. Hence, they can be a viable solution to this challenge.

In this work, we focus on building a constituency parser for Middle High German (MHG). MHG is a historical stage of the German language that was spoken between 1050 and 1350. It is the linguistic predecessor of Modern German (MG). Both languages have many similarities in word formation and grammatical features, e.g., similar word order patterns and inflectional systems (Salmons, 2018). The availability of MHG parse trees is extremely limited. The *Deutsche Diachrone Baumbank (German Diachronical Treebank, DDB)* (Hirschmann and Linde, 2023) comprises merely around 100 manually annotated parse trees, encompassing less than 3000 tokens. These resources are far from what is required to train an automatic syntactic analysis system, and are only suitable for use as test sets. On the other hand, there is an abundance of treebank resources available for MG, in particular the Tiger Treebank (Smith, 2003). Hence, we capitalize on the structural similarity between MHG and MG, as well as the rich MG treebank resources to develop a cross-lingual *delexicalized* constituency parsing model that we can directly apply to MHG sentences.

In the delexicalized approach, the parsing model operates on part-of-speech (POS) sequences rather than token sequences. We accomplish this by training a cross-lingual parser using POS sequences from high-resource source languages as input. Subsequently, we utilize this trained parser to directly parse POS sequences of low-resource target languages (McDonald et al., 2011).

In our work, we first train a delexicalized constituency parsing model on a delexicalized MG treebank. In order to parse MHG sentences with this model, we need to annotate them first with the POS tags used in the MG treebank. To this end, we train a POS tagger on an MHG corpus which has been manually annotated using a POS tag set similar, but not identical to the MG tag set. We employ a POS mapper to replace the MHG tags with the corresponding MG tags, ensuring the uniformity of the model's inputs across the two languages, which is a prerequisite of the delexicalization method. The experimental results show that our delexicalized constituency parser substantially outperforms all other zero-shot cross-lingual parsing baselines, achieving an F1-score of 67.3% on the MHG parse test set.

The delexicalization method is particularly well-suited for languages that (1) lack treebank resources, (2) possess sufficient annotated data for training POS taggers, and (3) exhibit syntactic similarities with a high-resource language. Our investigation of this realistic scenario shows the feasibility of automatic syntactic analysis for an ancient language.

4.3.2 Constituency Parsing

Neural Constituency Parsing Recent advances in constituency parsing have witnessed a growing emphasis on harnessing neural network representations, making a shift from the previously prominent role of grammars, whose relevance has gradually diminished. Cross and Huang (2016) propose a span-based constituency parsing system specifically designed to leverage the powerful representation capabilities of the bidirectional long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997). In this method, an input sentence is represented as a set of spans, and each span is assigned a score. The best-scoring parse tree is computed using dynamic programming techniques. They combine smaller spans into larger spans until the entire sentence is covered. Subsequently, several variations of the span-based method have been proposed, e.g. approaches replacing the inference algorithm with chart-based methods (Stern et al., 2017), using character-level representations instead of word-level representations (Gaddy et al., 2018), and replacing LSTMs with self-attention modules (Kitaev and Klein, 2018). Kitaev et al. (2019) take advantage of the newly developed pretrained language models (PLMs) and use BERT (Devlin et al., 2019) to compute the span representations, resulting in enhanced performance. Kitaev and Klein (2020) improve the runtime complexity of constituency parsing to linear time by reducing parsing to tagging.

Cross-Lingual Constituency Parsing There has been relatively limited scholarly attention dedicated to cross-lingual constituency parsing in recent studies, especially for target languages situated in low-resource settings, such as MHG. Kitaev et al. (2019) have employed the multilingual BERT model to train a single parser with parameters shared across languages. They jointly fine-tune the multilingual BERT on 10 languages utilizing a common BERT backbone, but the model contains distinct MLP span classifiers for each language to accommodate the different tree labels. However, their approach necessitates the availability of treebanks of all the encompassed languages as training datasets. Kaing et al. (2021) undertake a comprehensive series of experiments to validate the efficacy of delexicalization techniques for zero-shot cross-lingual constituency parsing. Additionally, their study underscores the significance of typological affinity in the source language selection. We build upon these investigations and apply their findings to the zero-shot parsing of MHG within a practical contextual framework.

Constituency Parsing on Historical German There is a notable scarcity of syntactically annotated corpora for historical German. In instances where annotated treebanks are absent, approaches such as rule-based, unsupervised, or zero-shot cross-lingual methods can be employed for constituency parsing. For instance, Chiarcos et al. (2018) have created a rule-based shallow parser for MHG. Recent advancements in the construction of such corpora encompass:

- *German Diachronical Treebank (DDB)*: a small yet syntactically deeply annotated corpus, comprising three subcorpora of different stages of German, i.e., Old High German, Middle High German, and Early New High German (Hirschmann and Linde, 2023). The construction of the DDB corpus is oriented towards the Tiger Corpus (Smith, 2003), one of the largest German treebanks.

- *UP Treebank of Early New High German (ENHG)*: a syntactically annotated corpus of ENHG containing 21,432 sentences consisting of 600,569 word tokens based on the Reference Corpus of ENHG (Demske, 2019).

- *Corpus of Historical Low German (CHLG)*: a Penn-style treebank of Middle Low German (Booth et al., 2020)

Contemporary work on historical German parsing based on previously mentioned corpora includes endeavors such as cross-dialectal parsing for ENHG based on CHLG (Sapp et al., 2023).

4.3.3 Languages and Corpora

The ancient language that we study in this work is Middle High German (MHG). MHG and Modern German (MG) are stages of the same Germanic language family, representing different historical periods. MHG emerged during the Middle Ages in the German-speaking regions of Central Europe. It was primarily used in literary and administrative contexts and played an important role in medieval literature, including epic poems such as the *Nibelungenlied* and *Minnesang* (courtly love poetry) (Salmons, 2018).

Linguistic Considerations of MHG MHG has a phonetic system that includes a set of vowel and consonant sounds. The pronunciation and sound patterns differ from those of MG, but some MHG words are still recognizable in MG. MHG has a more complex grammatical system, such as a more extensive case system with different noun and adjective declensions. Besides, verb conjugation has more intricate forms and patterns (Jones and Jones, 2019). In terms of orthography, the spelling and writing conventions of MHG are different from MG. For example, *ü*, the umlaut of *u*, is usually written *iu* in MHG. The transition from MHG to MG was a gradual process, occurring over several centuries. MG can be considered the linguistic descendant of MHG, with linguistic changes and developments shaping the language over time.

MHG Corpora Resources During the MHG period, the amount of textual material that survives to the present increases markedly. The *Reference Corpus of Middle High German* (*ReM*) (Klein et al., 2016) encompasses a large collection of non-literary and non-religious texts. ReM is a corpus of diplomatically transcribed and annotated texts of MHG with a size of around 2 million word forms. Texts in ReM have been digitized and richly annotated, e.g., with POS, morphological, and lemma features. The morphological annotation uses the HiTS tag set (Dipper et al., 2013), a tag set for historical German, derived from the Stuttgart-Tübinger Tag Set (STTS) for modern German texts (Schiller et al., 1995). Although the ReM corpus provides rich morphologically annotated text data for MHG, the availability of syntactically annotated data for MHG is severely limited, with only approximately 100 MHG parse trees included in the DDB treebank. In contrast, the treebank resources for MG are abundant. The Tiger Treebank (Brants et al., 2002), for instance, consists of approximately 40,000 sentences of German newspaper text, taken from the Frankfurter Rundschau.

4.3.4 Methods

In our work, we focus on developing a constituency parser for MHG. In the previous section, we reviewed annotated resources available for MHG and MG. Basically, we have ample treebank resources for MG and plenty of POS-tagged texts for MHG, whereas the treebank resources for MHG are extremely limited. Given the resource availability for MG and MHG along with the linguistic connection between the two languages, employing a cross-lingual constituency parsing approach utilizing delexicalization proves to be an effective solution. As Figure 4.7 shows, the delexicalized model is trained on the delexicalized inputs of MG. In the inference stage, the delexicalized parser is directly applied to MHG POS sequences. The delexicalization method requires that MHG and MG share the same set of POS tags. The final constituency parser for MHG (the right side of Figure 4.7 comprises three modules: (1) the delexicalized parser, (2) the MHG POS tagger, and (3) the POS mapper from MHG to MG. In the next section, we describe the delexicalized parsing system in more detail.

4.3.4.1 Delexicalized Parser

Our delexicalized MHG parser is based on the Berkeley neural parser (Benepar) (Kitaev and Klein, 2018), a span-based parser using self-attention. As illustrated in Figure 4.7, Benepar has an encoder-decoder architecture which combines a chart decoder with a sentence encoder based on self-attention. The sentence encoder computes contextualized representations for all word positions and combines them to form span representations. From the span representations, the parser computes label scores, which are subsequently used to incrementally construct a tree using a chart parsing algorithm (Sakai, 1961).

According to Kaing et al. (2021), Benepar exhibits two key features that are advantageous for cross-lingual transfer. Firstly, it employs a self-attentive encoder that effectively captures global context information and exhibits less sensitivity to word order. Secondly, the parser independently scores each span without considering the label decisions of its children or parent. This means that a failure in label prediction for a certain span does not strongly impact the label prediction for other spans (Gaddy et al., 2018). Consequently, the prediction errors resulting from local syntax variations between two languages have a limited effect on the overall prediction.

While our delexicalized parser adopts the same architecture as Benepar, there exist distinctions in the inputs of the two. Specifically, Benepar is trained on parse trees with words, whereas our delexicalized parser operates on POS sequences as inputs, i.e., tree strings devoid of words. Therefore, the delexicalized version of the MG treebank is required to train the delexicalized parser. For the MHG parsing in the inference, we feed the delexicalized model with the POS sequences of MHG sentences.

Delexicalization for MG We use the Tiger Treebank to train the delexicalized parsing model on MG parse trees. The parse trees in the Tiger Treebank contain additional semantic information, such as edge labels, and special structures, such as coreference indices and trace nodes. We remove all of them during delexicalization.

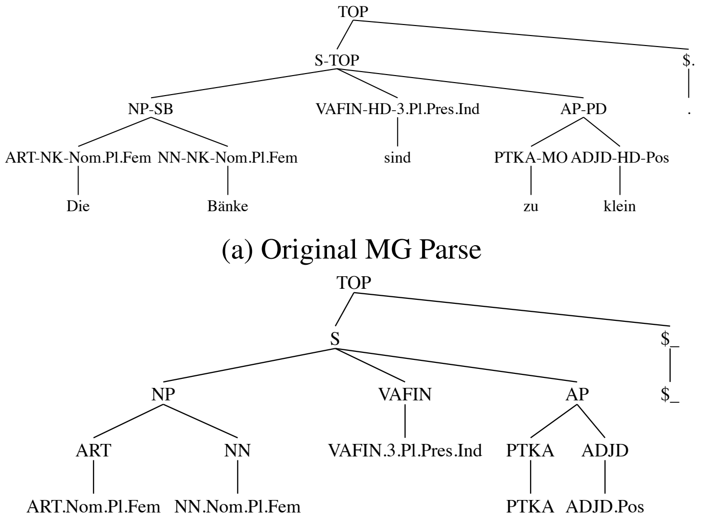
In the Tiger treebank, the label of each preterminal node contains not only the POS tag, but also morphological features, such as case, number, and gender. During delexicalization, we overwrite the word at the leaf node with this extended POS tag, but only keep the POS information in the label of the preterminal node. This means that the input of our delexicalized parser contains information about morphological features. Figure 4.8 shows an example of the delexicalization for an MG sentence. As shown the edge labels, e.g., "`NK`" are removed and the tokens are replaced by the POS tag combined with morphological features, e.g., "`ART.Nom.Pl.Fem`", where "`ART`" (determiner) is the POS tag, and "`Nom.Pl.Fem`" denotes the morphological information with case being nominative, number being plural, and gender being feminine.

4.3.4.2 Delexicalization for MG and MHG

MHG POS Tagger For the delexicalization of MHG sentences, we need a POS tagger for MHG. We use the RNNTagger of Schmid (2019) for this purpose, which annotates MHG sentences with POS tags as well as morphological features and has been trained on the ReM corpus. RNNTagger uses deep bidirectional LSTMs with character-based word representations.

4.3.4.3 Tag Set Mapping

The Tiger Treebank uses the STTS tag set, whereas the MHG version of the RNNTagger and the ReM corpus on which it was trained employ the HiTS tag set. Due to this discrepancy, we cannot directly use the POS labels from RNNTagger as input to the delexicalized parser. HiTS, for example, has separate tags for definite (`DDART`) and indefinite articles (`DIART`), whereas



(a) Original MG Parse

(b) Delexicalized MG Parse

Figure 4.8: An example illustrating the delexicalization process of an MG tree.

STTS uses the tag "`ART`" for both of them. Since the delexicalization method demands that the source and target languages share the same tag set, we have to map the MHG tags to the MG . The small MHG treebank that we use for evaluation purposes uses STTS and requires no mapping.

MHG Tag	**MG Tag**
CARDD	CARD
DDA	PDAT
DDART	ART
DIA	PIAT
DIART	ART
DID	PDAT
NA	NN
VAPS	ADJD.Pos

Table 4.14: Representative mapping pairs in the mapping dictionary.

The mapping process involves two dimensions. Firstly, we map the morphological features of MHG to those of MG. Secondly, we map the POS tags of MHG to those of MG primarily based on a mapping dictionary. Table 4.14 shows a selected part of the POS tag mapping dictionary. It should be noted that our mapping is not flawless due to certain challenges. For instance, the composite word in MHG "*enerde (on earth)*" is separated into "*auf*" and "*Erde*" in MG and are tagged as "`APPR|NA`". In the DDB treebank, such composite words are annotated with two separate tags combined with "|" in the DDB treebank. However, for simplification purposes, our mapping only retains the first part of the tag, leading to a loss of information.

4.3.5 Experiments

We begin by training Benepar on the delexicalized Tiger treebank for MG. Then we annotate the sentences of the small DDB treebank for MHG with RNNTagger and map the HiTS tags that it returns to STTS tags. Finally, we parse the POS tag sequences with the trained parser.

4.3.5.1 Datasets

In our experiments, we utilize the following three corpora (see also Table 4.15).

	Type	**Language**	**Size**	**Usage**
Tiger	Treebank	MG	50,474 trees	Parser training
DDB	Treebank	MHG	96 trees	Parser evaluation
ReM	POS-tagged corpus	MHG	2,269,738 tokens	POS tagger training

Table 4.15: Overview of the datasets.

Tiger Treebank The delexicalized parser is trained on the Tiger Treebank (Smith, 2003), which comprises a total of 50,474 parse trees for MG. We use a version of the Tiger Treebank which has been converted to the Penn Treebank format (Marcus et al., 1993). We delexicalize the Tiger corpus and divide it into a training set and a development set. The first 47,474 parse trees in the Tiger corpus comprise the training set, and the last 3,000 parse trees comprise the development set.

DDB The German Diachronic Treebank (DDB) (Hirschmann and Linde, 2023) consists of a limited number of 100 parse trees for MHG. Due to the small data size, we utilize the DDB treebank solely for the cross-lingual evaluation of the delexicalized parser. To prepare the DDB treebank for evaluation, we perform preprocessing steps, including converting it to the format of the Penn Treebank and removing incomplete parse trees and parse trees with mostly Latin words. We also removed numbers and periods that formed the first token of a parse tree, and corrected a few more minor problems. At the end, we had 96 sentences for evaluation purposes.

ReM The Reference Corpus for Middle High German (ReM) (Klein et al., 2016) is an extensive collection of texts written in MHG. This corpus encompasses approximately 2.3 million tokens and provides comprehensive linguistic annotations, including POS tags, morphological analysis, lemma features, and more. The ReM corpus has been used by Schmid (2019) to train the MHG version of his RNNTagger, which annotates MHG texts with POS tags and morphological features.

4.3.5.2 Baselines

We evaluate the performance of our proposed delexicalized MHG parser, which is based on the Benepar parser (Kitaev and Klein, 2018), and compare it with the cross-lingual transfer performance of the original Benepar without using the delexicalization method and other parsing approaches that incorporate pretrained language models, which have shown promising results in various NLP tasks.

Vanilla Benepar The vanilla Benepar model is trained directly on the original training set of the Tiger Treebank for MG without delexicalization. After training, the parser is directly used to parse the MHG sentences as token sequences. This allows us to compare the performance of the delexicalized MHG parser with the vanilla Benepar model, highlighting the impact of delexicalization on cross-lingual parsing performance.

Tetra-Tagging with PLMs Tetra-tagging (Kitaev and Klein, 2020) is a technique for reducing constituency parsing to sequence labeling. In this approach, special parsing tags are predicted in parallel using a PLM and then merged into a parse tree. In our experiment, we use the pretrained German BERT model (Chan et al., 2020) and the multilingual BERT model (Devlin et al., 2019) available on the HuggingFace website (Wolf et al., 2020). We start by fine-tuning these models on the Tiger Treebank using the Tetra-tagging technique. Subsequently, we evaluate their performance on the MHG parse test set.

4.3.5.3 Evaluation

Following Kitaev and Klein (2018), we use the standard `evalb` measures (Sekine and Collins, 1997; Collins, 1997) for the parser quality evaluation. `evalb` is a software tool that provides metrics to assess the accuracy and similarity of parsed sentences against reference or gold standard parse trees, including precision, recall, F1 score, and complete match.

- **Precision** measures the proportion of predicted constituents in the generated parse tree that are also contained in the reference parse tree. It quantifies the accuracy of the parser in correctly identifying constituents.

- **Recall** measures the proportion of constituents in the reference parse tree that were predicted by the parser in the generated parse tree. It quantifies the parser's ability to generate all the constituents present in the reference parse tree.

- **F1 Score** is the harmonic mean of precision and recall.

- **Complete Match** measures the proportion of predicted parse trees that were exactly identical to the respective reference parse trees.

As is the standard practice, the evaluation disregards POS labels and punctuation.

4.3.5.4 Training Setup

For training the delexicalized parser, we adopt the same hyperparameter settings as described in (Kitaev and Klein, 2018). The encoder architecture consists of a character-level bidirectional

LSTM neural network. The size of the feedforward layer is set to 2048, and the character embedding dimension is 64. The batch size is set to 32, the learning rate is 5e-5, and the maximum sequence length of the encoder is 512. We use the random seed 10 for training. We conduct all our experiments using a server with 8 GPUs with 11GB RAM (NVIDIA GeForce GTX 1080 Ti).

4.3.6 Results and Analysis

4.3.6.1 Main Results

	Recall		**Precision**		**FScore**		**CM**	
	MG	MHG	MG	MHG	MG	MHG	MG	MHG
Baselines								
Vanilla Benepar	84.18	34.41	87.57	44.40	85.84	38.77	45.80	0.00
Tetra-gBERT	**86.31**	23.20	**88.19**	29.53	**87.24**	25.98	**51.70**	3.12
Tetra-mBERT	60.68	19.69	65.61	23.25	63.15	21.32	21.35	0.00
Our proposed method								
Dexparser	81.39	**64.72**	84.89	**70.19**	83.10	**67.34**	39.03	**12.50**

Table 4.16: Main results of the cross-lingual parsing transfer performance of different parsers. **CM** refers to “complete match”. gBERT refers to the pretrained German BERT, and mBERT refers to the multilingual version of BERT. The best value of each column is indicated in **bold**.

Table 4.16 shows the parsing performance of different cross-lingual parsers. Notably, our proposed parser attains the highest scores across all metrics for MHG, demonstrating that the delexicalized parser possesses superior cross-lingual parsing performance on MHG. Our delexicalized parser demonstrates substantial advantages in parsing MHG, achieving an impressive increase of almost 30% points in F1 score. Besides, it achieves comparable results on MG. In terms of the baselines, the Vanilla Benepar and the Tetra-gBERT parser both achieve relatively high recall and precision for MG but have noticeably lower values for MHG. The Tetra-mBERT parser exhibits lower values for both recall and precision for both MG and MHG. It is worth noting that the parsing performance of the delexicalized model on the source language MG is surpassed by the two strong baselines, Vanilla Benepar and Tetra-gBERT. This outcome is expected as the delexicalization process diminishes the semantic information present in the input sequences. However, the trade-off of the performance loss in MG leads to a big leap in the cross-lingual parsing performance for MHG.

Our delexicalized constituency parser exhibits outstanding performance on the MHG test set, attaining an impressive F1-score of 67.3%. This substantial improvement outperforms the best zero-shot cross-lingual baseline by a considerable margin of 28.6%. Although there is a slight decline in the parsing performance for MG, the trade-off proves worthwhile considering the substantial gains achieved in parsing MHG. This emphasizes the effectiveness of the delexicalized approach in facilitating cross-lingual transfer and highlights its potential for parsing ancient and historical languages like MHG.

4.3.6.2 Ablation Study

	Recall	**Precision**	**FScore**	**CM**
Delexicalized parser using gold tags	**66.18**	**71.17**	**68.59**	**14.58**
- *using predicted tags*	64.72	70.19	67.34	12.50
- *without mapping*	59.16	68.82	63.63	7.29
- *without morphological information*	48.66	65.38	55.8	9.28

Table 4.17: The MHG parsing results with the delexicalized parser in the ablation study.

We now examine how the parsing performance changes (i) as we replace predicted POS tags with gold-standard POS tags, (ii) as we use the original HiTS tags instead of mapping them to STTS tags, and (iii) as we remove the morphological features from the parser input. Table 4.17 presents the results of our ablation study.

Goldstandard POS Tags We observe that the f-score of the delexicalized parser increases by 1.3% points when it processes gold standard POS tag sequences instead of POS tag sequences predicted by RNNTagger. This finding underscores the quality of the POS tags predicted by RNNTagger. We loose very little performance due to POS tagging errors.

Tag Set Mapping Table 4.17 demonstrates a noticeable decline in parsing performance from 67.34% to 43.43% in terms of F1 score when the delexicalized MHG sequences are directly processed by the cross-lingual parser without mapping them from HiTS to STTS. This finding highlights the indispensability of mapping from MHG to MG for maintaining satisfactory parsing performance. The results underscore the significance of aligning the tag sets between MHG and MG to ensure effective cross-lingual parsing and emphasize the necessity of this mapping process in our approach.

Morphological Information The inclusion of morphological markers provides the neural model with valuable additional information for parsing MHG sentences. In our experiments, we augment the delexicalized MHG sequences with morphological information, such as case, gender, number, and more. The outcomes of the ablation study clearly indicate that removing this morphological information from the delexicalized input sequences obviously impairs parsing performance. Specifically, this exclusion leads to a noticeable decline in the F1 score, amounting to a reduction of 11.5%.

4.3.6.3 Case Study

Figure 4.9 shows two MHG trees generated by our delexicalized parser and the corresponding gold standard trees for comparison. This case study reveals that the delexicalized parser demonstrates relatively accurate predictions of constituents when compared to the reference trees, especially for short MHG sentences. Some prediction errors in constituents stem from the intricacy and the ambiguity of the MHG grammar, as exemplified by the case of "*her*" in Example 2. From

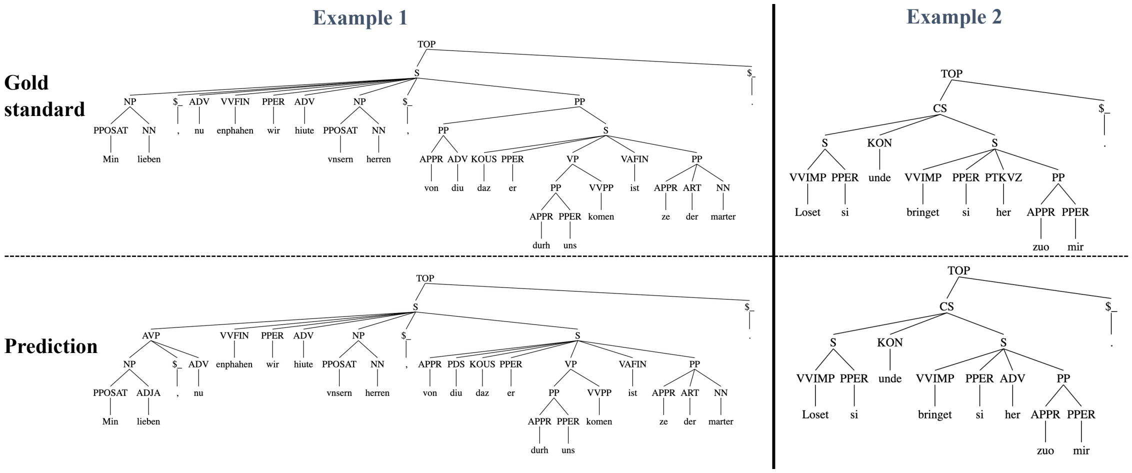

Example 1
Example 2
Gold standard
TOP
S
$_
NP $_ ADV VVFIN PPER ADV NP $_ PP
PPOSAT NN , nu enphahen wir hiute PPOSAT NN , PP S
Min lieben vnsern herren
APPR ADV KOUS PPER VP VAFIN PP
von diu daz er PP VVPP ist APPR ART NN
APPR PPER komen ze der marter
durh uns
.
TOP
CS $_
S KON S .
VVIMP PPER unde VVIMP PPER PTKVZ PP
Loset si bringet si her APPR PPER
zuo mir
Prediction
TOP
S
$_
AVP VVFIN PPER ADV NP $_ S
NP $_ ADV enphahen wir hiute PPOSAT NN , APPR PDS KOUS PPER VP VAFIN PP
PPOSAT ADJA , nu vnsern herren von diu daz er PP VVPP ist APPR ART NN
Min lieben
APPR PPER komen ze der marter
durh uns
.
TOP
CS $_
S KON S .
VVIMP PPER unde VVIMP PPER ADV PP
Loset si bringet si her APPR PPER
zuo mir

Figure 4.9: Two examples of the trees generated by our delexicalized parser compared to the reference parses.

a linguistic perspective, determining whether "*her*" functions as an adverb (`ADV`) or a separated verb prefix (`PTKVZ`) poses challenges. However, in longer and more complex sentences, e.g., the sentence in Example 1, the parser typically maintains a high level of accuracy locally while occasionally struggling to accurately determine the overall structure of the entire sentence. Besides, the presence of noise in the ancient texts is another factor that can impact the effectiveness of the cross-lingual parsing for MHG. Overall, the qualitative analysis provides further evidence of the effectiveness of the delexicalized parser for MHG, emphasizing its ability to accurately predict constituents, especially in shorter sentences. While challenges may arise in handling longer and more complex sentences, the delexicalized parser showcases promising results, contributing to the advancement of MHG parsing.

4.3.7 Sum-Up

In summary, our study presents an effective cross-lingual constituency parsing approach for ancient languages, specifically focusing on the parsing of Middle High German (MHG) sentences. Through the utilization of delexicalization and the similarities between MHG and Modern German (MG), we have developed a delexicalized parser based on the rich treebank resources of MG, which demonstrates remarkable performance in parsing MHG sentences. Our experimental results showcase the efficacy of the delexicalized approach, outperforming existing baselines and achieving substantial improvements in parsing accuracy. These findings highlight the practicality and promise of our approach for parsing historical and ancient languages, addressing the challenges posed by limited annotated data and linguistic variations.

Chapter 5

Efficient NLP Methods for Low-Resource Settings

Summary of This Chapter

While much of the recent progress in NLP has been driven by LLMs trained on vast datasets with substantial computational resources, real-world applications often face significant resource constraints. These constraints manifest not only in the form of **low-resource languages**, as discussed in the previous two chapters (Chapter 3 and Chapter 4), but also in practical scenarios where both annotated data and computational capacity are limited, i.e., **low-resource data and computing** settings. As NLP systems are increasingly deployed in diverse domains and languages, the need for methods that can operate efficiently under such constraints has become ever more pressing. Addressing these challenges requires strategies that can maximize the utility of available data and minimize the computational burden of model adaptation. Two prominent approaches have emerged in this context: data augmentation, which seeks to expand and diversify training data in low-resource settings, and parameter-efficient fine-tuning, which aims to adapt large models to new tasks or domains by updating only a small subset of parameters. Both approaches are crucial for enabling scalable, inclusive, and sustainable NLP in scenarios where traditional full-model fine-tuning or large-scale data collection is infeasible.

This chapter investigates efficient NLP methods tailored for low-resource settings, focusing on both data scarcity and computational limitations. We present two complementary contributions, each addressing a key aspect of the efficiency challenge.

First, we tackle the problem of low-resource multi-domain dialogue generation by proposing a unified data augmentation framework, AMD^{2}G. This method systematically decouples domain-agnostic and domain-specific features through a de-domaining process, enabling models to learn shared expressive patterns across domains before adapting to the unique characteristics of the target domain. Extensive experiments on Chinese dialogue datasets spanning five domains demonstrate that AMD^{2}G consistently outperforms both direct domain-specific training and naive multi-domain training, highlighting its effectiveness in leveraging cross-domain knowledge for data-scarce applications (§5.1).

Second, we address the challenge of computational resource constraints in model adaptation by introducing a novel parameter-efficient fine-tuning approach, GNNavi. Inspired by recent advances in understanding information flow in in-context learning, GNNavi integrates a graph neural network (GNN) layer into the deep layers of large language models. This design explicitly guides the aggregation and distribution of information within prompts, allowing for effective adaptation by updating only a small fraction of model parameters. Our experiments on few-shot text classification tasks with GPT-2 and Llama2 demonstrate that GNNAVI achieves superior performance and training efficiency compared to established parameter-efficient methods such as LoRA, Prefix-Tuning, and Adapters (§5.2).

By advancing both data-centric and model-centric efficiency techniques, this chapter contributes to the broader goal of making NLP technologies more accessible, adaptable, and sustainable across a wide range of languages and domains. These methods not only address immediate practical bottlenecks in low-resource scenarios, but also lay the groundwork for future research on scalable and inclusive language technologies.

5.1 Data Augmentation for Low-Resource Multi-Domain Dialogue Generation

This section corresponds to the following work:

> Yongkang Liu*, **Ercong Nie***, Zheng Hua, Zifeng Ding, Daling Wang, Yifei Zhang, Hinrich Schütze. 2024. A Unified Data Augmentation Framework for Low-Resource Multi-Domain Dialogue Generation. In Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD 2024). Springer.
> * equal contributions.

Declaration of Co-Authorship. Yongkang Liu proposed the idea of using de-domaining data for data augmentation to enhance the performance of low-resource multi-domain dialogue generation tasks. I conceived the framework of AMD^{2}G based on the idea of de-domaining. Yongkang Liu dealt with the data processing, and Zheng Hua ran most of the experiments. I summarized the experimental results and drafted the paper. Zifeng Ding attended multiple rounds of discussions and provided valuable feedback. Yifei Zhang and Hinrich Schütze are supervisors of this project.

Summary of This Section

Current state-of-the-art dialogue systems heavily rely on extensive training datasets. However, challenges arise in domains where domain-specific training datasets are insufficient or entirely absent. To tackle this challenge, we propose a novel data **A**ugmentation framework for **M**ulti-**D**omain **D**ialogue **G**eneration, referred to as **AMD^{2}G**. The AMD^{2}G framework consists of a data augmentation process and a two-stage training approach: domain-agnostic training and domain adaptation training. We posit that domain corpora are a blend of domain-agnostic and domain-specific features, with certain representation patterns shared among diverse domains. Domain-agnostic training aims to enable models to learn these common expressive patterns. To construct domain-agnostic dialogue corpora, we employ a ***de-domaining*** data processing technique used to remove domain-specific features. By mitigating the effects of domain-specific features, the model trained on the de-domained corpora can effectively learn common expression patterns in different domains. Subsequently, we adapt the learned domain-agnostic features to the target domain through domain adaptation training. We conduct experiments on Chinese dialogue datasets from five different domains and show that AMD^{2}G achieves superior performance compared to both direct training on the target domain corpus and collective training on all five domain corpora. Our work underscores AMD^{2}G as a viable alternative solution for low-resource multi-domain dialogue generation.

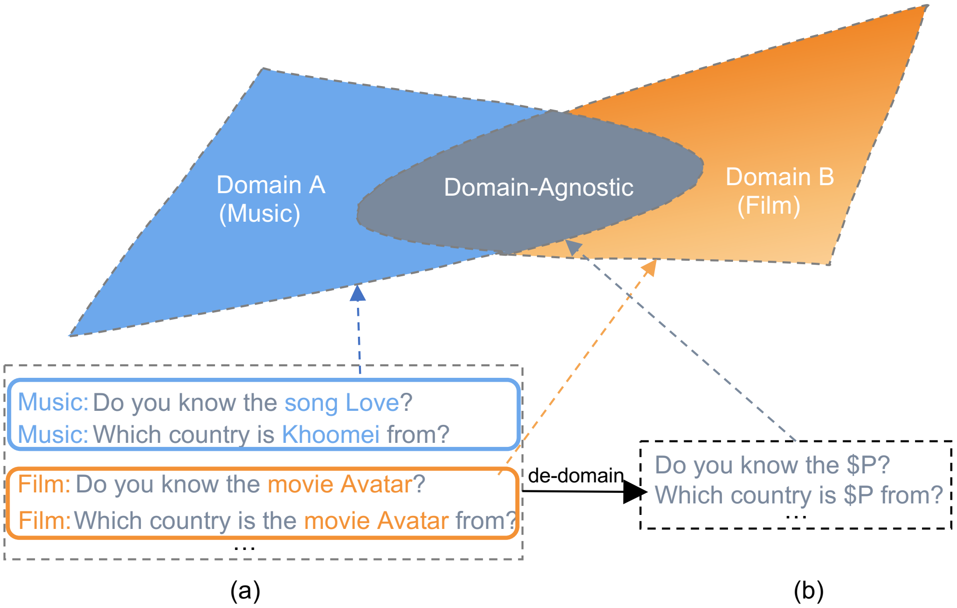

Domain A
(Music)
Domain-Agnostic
Domain B
(Film)
Music: Do you know the song Love?
Music: Which country is Khoomei from?
Film: Do you know the movie Avatar?
Film: Which country is the movie Avatar from?
...
de-domain
Do you know the $P?
Which country is $P from?
...
(a)
(b)

Figure 5.1: Illustration of corpus composition in different domains. (a) represents domain-specific corpora, (b) stands for domain-independent corpora. The overlap of Domain A (blue) and Domain B (Orange) represents domain-agnostic data, while non-overlapping regions signify domain-specific data.

5.1.1 Background and Motivation

The efficacy of established sequence-to-sequence methodologies in constructing dialogue systems has demonstrated remarkable success in previous research (Serban et al., 2016; Liu et al., 2023c, 2022d; Li et al., 2023b). More recently, the notable achievements of Large Language Models (LLMs), including Blender (Roller et al., 2021), Meena (Adiwardana et al., 2020), ChatGPT (Ouyang et al., 2022), and GPT-4 (OpenAI, 2023), have prompted the research community to increasingly embrace generative models as the go-to approach. However, training these models often requires huge corpora. Unfortunately, the availability of domain-specific corpora remains notably limited across many domains, including medicine, finance, and military, due to concerns on security, copyright, and other constraints (Hathaliya and Tanwar, 2020). On the other hand, while the current LLMs exhibit excellent comprehension and generation capabilities, they often suffer from hallucinations in tasks with obvious domain characteristics and strong factuality (Bang et al., 2023b; Liu et al., 2023b; Ji et al., 2023; Yuan et al., 2024), particularly in low- and zero-resource contexts. Hence, devising strategies to leverage corpora from disparate domains to enhance the performance and accuracy of the target domain remains a relevant pursuit in the era of LLMs.

Existing methods for cross-domain dialogue generation can be categorized into three groups (Qin et al., 2020): i) Separate Pattern (Wen et al., 2018; Qin et al., 2019; Wu et al., 2021; Li et al., 2023c); ii) Mixed Pattern (Madotto et al., 2018; Wu et al., 2019b; Lin et al., 2021; Kim et al., 2023; Yang et al., 2023a; Ma et al., 2023b); and iii) Shared-Private Pattern (Zhong et al., 2018; Chen and Cardie, 2018; Wu et al., 2019c,a; Bang et al., 2023a). The Separate Pattern involves training the model separately for each domain, necessitating an adequate corpus for each domain. In contrast, the Mixed Pattern combines multi-domain datasets by prioritizing domain-agnostic features while disregarding domain-specific ones. However, models based on the Mixed Pattern may struggle to capture features from low-resource domains compared to high-resource ones. The Shared-Private Pattern extracts domain-agnostic and domain-specific features using shared and private modules, respectively, also relying on sufficient domain corpora. While these methods have demonstrated promising results, they all presuppose the availability of ample corpus data.

In the context of dialogue generation within low-resource domains, the effective leverage of resources from other domains holds significant importance. As illustrated in Figure 5.1, the domain corpora encompass both domain-agnostic and domain-specific information. For instance, in the film domain, the expression *"Do you know the movie Avatar"* domain comprises domain-specific information *"movie Avatar"* alongside the domain-agnostic fragment *"Do you know the ..."*. Similarly, phrases like *"song Love"* and *"Khoomei"* carry domain-specific features of the music field. Notably, disparate domains exhibit shared expression patterns within their corpora. Leveraging these shared features provides an opportunity to enhance the performance of dialogue generation in low-resource domains.

Accordingly, we propose a simple yet effective data **A**ugmentation method for **M**ulti-**D**omain **D**ialogue **G**eneration, termed **AMD^{2}G**. As shown in Figure 5.2, we initially build a domain dictionary for each domain automatically. Subsequently, the domain corpora undergo de-domaining through the usage of domain dictionaries. Specifically, placeholders replace domain-specific

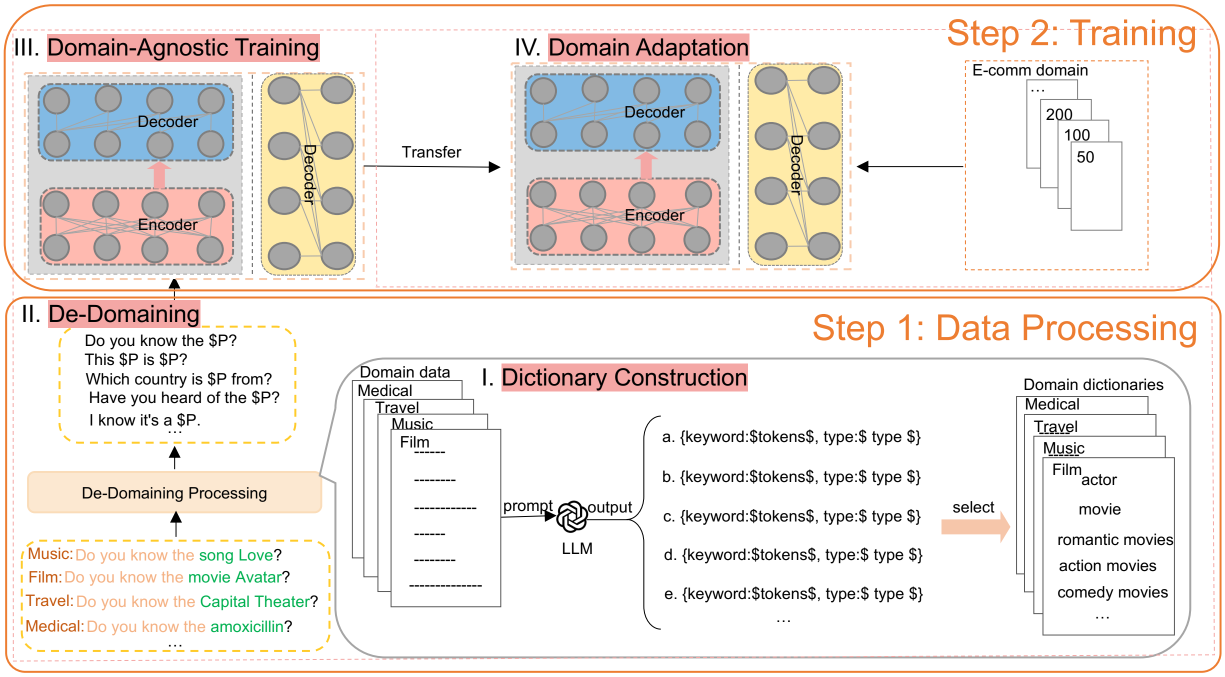

Step 2: Training
III. Domain-Agnostic Training
Decoder
Encoder
Decoder
Transfer
IV. Domain Adaptation
Decoder
Encoder
Decoder
E-comm domain
…
200
100
50
II. De-Domaining
Step 1: Data Processing
Do you know the $P?
This $P is $P?
Which country is $P from?
Have you heard of the $P?
I know it's a $P.
…
De-Domaining Processing
Music: Do you know the song Love?
Film: Do you know the movie Avatar?
Travel: Do you know the Capital Theater?
Medical: Do you know the amoxicillin?
…
Domain data
Medical
Travel
Music
Film
I. Dictionary Construction
prompt
output
LLM
a. {keyword:$tokens$, type:$ type $}
b. {keyword:$tokens$, type:$ type $}
c. {keyword:$tokens$, type:$ type $}
d. {keyword:$tokens$, type:$ type $}
e. {keyword:$tokens$, type:$ type $}
…
select
Domain dictionaries
Medical
Travel
Music
Film
actor
movie
romantic movies
action movies
comedy movies
…

Figure 5.2: Schematic diagram of **AMD^{2}G** framework. The target domain is E-Commerce, and the domains used for de-domaining are Film, Music, Travel, and Medical. $\$P$ represents the placeholder. The method supports both **encoder-decoder** and **decoder-only** structures.

keywords in the corpus identified in the domain dictionary. The models are then fine-tuned on the combined de-domained corpora to learn common representation patterns across different domains. Finally, low-resource fine-tuning is conducted on the target domain dataset to acquire domain-specific features. We conduct experiments with AMD^{2}G using Chinese conversation datasets from five distinct domains. This choice is motivated by the morphological simplicity of the Chinese language, characterized by few inflectional variations, aligning well with our de-domaining processing at the lexical level. We compare our proposed method with two baselines: direct training on the target domain corpus and training on all five domain corpora collectively. Our experimental findings demonstrate that the integration of AMD^{2}G consistently enhances model performance across all five domains.

In summary, our contributions are as follows:

- We introduce ***de-domaining*** for multi-domain dialogue generation datasets, a data augmentation technique that effectively reduces the impact of domain-specific features by extracting shared representations across domains.

- We propose **AMD^{2}G**, a simple yet effective alternative framework for the multi-domain dialogue generation task in low-resource settings.

- We conduct experiments with AMD^{2}G across five domains, demonstrating its superiority over direct training in the target domain and joint training across all five domains.

5.1.2 Multi-Domain Dialogue Generation

Dialogue systems are used as intelligent agents in various domains due to their ability to generate fluent and natural responses. Cross-domain learning refers to the technology of transferring knowledge from other domains to the target domain. The initial approach is to blend all domain corpora to learn domain-independent features. One of the drawbacks of this approach is the lack of domain-specific knowledge. Wu et al. (2019b) propose to use a global-to-local pointer mechanism search technique for external knowledge to enhance the domain-specific knowledge awareness of models. He et al. (2020) employs pairwise similarity to distill contextually unrelated KB records to improve the quality of domain knowledge. Xie et al. (2022) integrates domain-specific knowledge in the form of text-to-text format based on T5 (Raffel et al., 2020). Ma et al. (2023b) propose a domain attention module with distributional signatures of the dialogue corpus to capture domain-specific knowledge. These methods may also lead to a long-tail distribution of domain data, making models trained severely biased under low-resource conditions. Another line of research is to train separate models for each domain, focusing on domain-specific features. Madotto et al. (2018) learns domain-specific features by combining external knowledge through a memory network (Sukhbaatar et al., 2015). Qin et al. (2019) proposes to employ two-step retrieval and attention mechanisms to improve the quality of domain-specific features. Wu et al. (2021) proposes to continue pre-training on the domain corpus to adapt the language model to a specific domain. The shared-private framework, which combines the advantages of the above two, is a better choice. Zhong et al. (2018) uses global modules to share parameters and local modules to learn domain-specific features. Wu et al. (2019c) allows domain-specific features to interact through a shared-private mechanism. Bang et al. (2023a) learns task-related features by adding adapters for each task.

5.1.3 Methodology

As depicted in Figure 5.2, AMD^{2}G primarily comprises two steps: data processing and training. In data processing, the domain corpus is de-domained by using the constructed domain dictionaries. In the training step, domain-agnostic training first allows models to learn shared patterns among multiple domains, while the domain adaptation phase enables models to capture domain-specific features.

5.1.3.1 Problem Formulation

An instance for one domain can be represented as (C, R), where C=$\{u_1, u_2, ..., u_n\}$ with n utterances representing the context of the dialogue. Here, u_i represents the i-th utterance, and R represents the corresponding response. Our goal is to build a corresponding dialogue generation system $P(R|C)$ using corpora from other domains under the low-resource condition in the target domain. Please refer to the experimental settings for detailed settings of low resources (i.e., Section 5.1.4).

Note that AMD^{2}G can be applied to models of both encoder-decoder and decoder-only structures. For models with an encoder-decoder structure, the encoder is responsible for encoding the dialogue history C, and the decoder generates responses R based on the encoded representation. For models with a decoder-only structure, we concatenate the dialogue history C and responses R into a consecutive sequence and perform sequence modeling by autoregression.

5.1.3.2 De-Domaining Data Processing

We observe that data from different domains exhibits shared representation patterns. Through de-domaining operations, corpora from different domains can be transformed into a unified space devoid of domain-specific features. This process mitigates the influence of domain-specific features, facilitating the learning of domain-independent features by models.

De-Domaining Domain-specific corpora are de-domained based on the usage of domain dictionaries. Specifically, we replace all words or phrases present in the domain dictionary with designated placeholders, wherein all tokens within an involved phrase are substituted by a single placeholder. As a result of the de-domaining process, the domain-specific data no longer retains its specific characteristics.

Dictionary Construction We combine LLM-extracted terms with existing term banks to construct high-quality domain dictionaries tailored to specific domains. Initially, we employ TechGPT (Ren et al., 2023), a Chinese LLM, to extract domain entities as keywords. TechGPT has been enhanced for various information extraction tasks through the integration of domain knowledge graphs (Ren et al., 2018) facilitated by BELLE (Yunjie et al., 2023), which is specialized in extracting domain keywords. We apply the following prompt to TechGPT to extract entities from the domain corpora. `$Context$` represents the context composed of dialogue history and response. `$Domain$` represents the domain name.

> **Prompt:** `文本是$Context$。领域是$Domain$。请输出$Domain$关键词。`
> **Translation:** `The context is $Context$. The domain is $Domain$. Please output keywords related to $Domain$.`
> `($Domain$ ∈ {Film, Music, Travel, Medical, E-commerce})`

In addition to keywords extracted by the LLM, we utilize existing terminology banks offered by Chinese input method providers, including QQPinyin[1], Baidu[2], and SougouPinyin[3]. We retrieve the terms specific to each domain from these sources and merge them into the respective domain's dictionary Table 5.1 presents the statistical overview of the dictionary, including its size, coverage ratio concerning the training set, and the number of replaced tokens. Notably, the domain dictionary's coverage rate for the training corpus exceeds 95%, with a significant reduction in domain-specific terms.

[1] http://cdict.qq.pinyin.cn/v1
[2] https://shurufa.baidu.com/dict
[3] https://pinyin.sogou.com/dict/

Domain	#Keyword	#Cov	#RToken
Film	2463	100%	38680
Music	1427	100%	32305
Travel	1006	100%	33820
Medical	18749	99.80%	37065
E-comm	1384	95.10%	19602

Table 5.1: Statistic overview of domain dictionaries. *#Keyword* represents the number of keywords in the dictionary. *#Cov* represents the proportion of training set examples covered by the dictionary. *#RToken* represents the number of domain words removed from the training set.

5.1.3.3 Domain-Agnostic Training and Domain Adaptation

We conduct the first stage of fine-tuning on a mixed domain-agnostic corpus, excluding the target domain, to learn domain-independent features. Models pay more attention to domain-independent features in the domain-agnostic training phase. The purpose of domain adaptation is to transfer domain-independent knowledge to the target domain, allowing models to learn domain-specific features. Subsequently, models are initialized with the weights from the domain-agnostic training stage and then fine-tuned on the low-resource target domain corpus.

5.1.3.4 Domain Similarity

We believe that the similarity between domains is an important factor for AMD^{2}G. To explore the impact of different domains, we propose a simple domain similarity evaluation method. Specifically, we employ n-gram recall between domains as the similarity metric. Given that the n-gram sets of domains A and B are A_n and B_n respectively, the similarity score of domain A relative to domain B is:

$$\textbf{Similarity}_{A2B} = \frac{\sum A_n \cap B_n}{\sum B_n} \tag{5.1}$$

The similarity of domain B to A $\textbf{Similarity}_{B2A}$ can be computed in the same way. $\textbf{Similarity}_{*2B}$ represents the degree of similarity between other domains and domain B. Theoretically, the greater the similarity between the mixed data and target domain, the greater the benefits of data augmentation to the target domain. Table 5.2 shows the average similarity scores between domains. Note that all results are based on the training set after removing domain words. Expression paradigms usually consist of more than two words, so similarity scores based on 2-gram and above can better show the degree of similarity between domains. An elevated similarity score signifies a greater overlap of paradigms between the target domain and other domains.

5.1.4 Experiments

Datasets In this paper, we experiment on Chinese dialogue generation datasets from five domains (i.e., Film, Music, Travel, Medical, and E-commerce). Film, Music, and Travel are all from KdConv (Zhou et al., 2020) datasets. KdConv is a Chinese multi-domain conversation

Domain	Uni	Bi	Tri	Quad
O2Music	77.67	32.41	15.73	7.80
O2Travel	67.01	25.42	12.30	5.62
O2Film	63.91	23.30	11.13	5.41
O2Ecomm	80.95	28.80	9.43	2.26
O2Medical	63.53	16.15	5.55	1.86

Table 5.2: Similarity scores between different domains. O2*$Domain$* represents the average similarity score of other domains to *$Domain$*. *Uni*, *Bi*, *Tri*, and *Quad* represent the recall rates of 1-gram, 2-gram, 3-gram, and 4-gram respectively. All values are magnified by a factor of 100.

dataset comprising 4.5K conversations from three domains: Film, Music, and Travel. It contains 86K dialogue with 19.0 turns on average. These conversations feature in-depth discussions and natural transitions between multiple topics. The **Film**, **Music**, and **Travel** domains contain 1,500 training samples, 150 validation samples, and 150 test samples, respectively. **MedDG** is a large-scale Chinese medical dialogue dataset, which contains 14,864 training samples, 2,000 validation samples, and 1,000 test samples, respectively (Liu et al., 2022b). **E-commerce** is a large-scale e-commerce conversation dialogue dataset, containing 500,000 positive training examples, 1,000 validation examples, and 1,000 test examples, respectively (Zhang et al., 2018). Similarly, we randomly selected 2,000 examples as the training set. In order to meet the low-resource setting, we extract 2000 conversations as the training set for **MedDG** and **E-commerce**. It is worth noting that we do not use the knowledge base.

Models To evaluate the effectiveness and robustness of our proposed method, we apply **AMD^{2}G** to different types of models, including encoder-decoder and decoder-only structures. For the encoder-decoder structure models, we employ the basic **Transformer** (Vaswani et al., 2017) structure as well as Chinese pre-trained language models such as **CPT** (Shao et al., 2021) and the Chinese version of **BART**[4] (Lewis et al., 2020). The decoder-only structure model used in our experiment is the Chinese version of the pre-trained model **GPT-2**[5] (Radford et al., 2019).

Baselines For each model, we first compare AMD^{2}G with training only on the original domain training set and training on the mixed training set of all domains. Besides, we compared AMD^{2}G with two other multi-domain transfer methods: **TS-NET** (Peng et al., 2019) and **DA-NET** (Ma et al., 2023b). TS-NET uses a teacher-student network mechanism to transfer knowledge from other domains to the target domain, while DA-NET uses domain attention to realize knowledge transfer. We apply different methods to two types of models. One is the pre-trained Chinese model, **BART**, and the other is a language model that adopts the architecture of gated recurrent neural networks (GRU), a variant of RNN (Chung et al., 2014).

[4] https://huggingface.co/fnlp/bart-base-chinese

[5] https://huggingface.co/uer/gpt2-chinese-cluecorpussmall

Implementation Details We implement our model and baselines using the Huggingface library (Wolf et al., 2020) and train baselines on a server with RTX A6000 GPUs (48 GB). We consider at most 10 turns of dialogue context and 50 words for each utterance. The batch size is 32, the minimum decoding length is set to 10, the maximum decoding length is set to 128, the warmup steps are 1,000, and the initial learning rate is 5e-5. We use the AdamW (Loshchilov and Hutter, 2017) optimizer to update model parameters. The beam size and length penalty coefficient are set to 6 and 1.0, respectively. The random seed of the sampled data is set to 12345678. The random seed of the training process is set to 12345. We use GLoVe (Pennington et al., 2014) to train 300-dimensional word vectors based on the training set for evaluation. The values of hyperparameters described above have been optimized on the validation set. We explore low-resource scenarios using 5%, 10%, 20%, 30%, and 40% of the target corpus for training.

Evaluation metrics Following previous studies (Liu et al., 2022d; Li et al., 2017; Xu et al., 2018; Liu et al., 2023c), we use both automatic and human evaluations to assess the performance of models. Automatic evaluations include BLEU, Rouge, Dist, and Embedded metrics. Perplexity is also measured as an additional metric. For human evaluation, we request annotators to score the generated responses with respect to three aspects: fluency, diversity, and relevance. Each dimension is divided into three levels: 0, 1, and 2. In terms of fluency, 0 means no fluency, 1 means average fluency, and 2 means high fluency. Other evaluation dimensions are similar to fluency. After collecting the assessments from annotators, the final score is the average of all samples. Note that we use an improved version of BLEU (Yang et al., 2018) that is more in line with human evaluation, and the calculated score will be lower than the original BLEU (Papineni et al., 2002).

Model	Corpus	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Rouge-L	Dist-1	Dist-2	Embed A/E/G			AVE↑	PPL↓
Film													
Transformer	target	23.68	13.66	9.41	6.65	31.20	17.11	38.69	74.36	56.67	84.43	35.59	6.7480
	mix	22.60	12.72	8.51	6.07	29.91	20.16	48.86	77.47	58.19	86.69	37.12	6.4009
	AMD^{2}G	29.67	16.96	11.92	8.70	33.52	22.92	51.18	78.64	59.94	86.87	**40.03**	**6.2826**
CPT	target	26.18	14.65	10.05	7.24	31.32	25.34	56.69	78.07	58.81	86.67	39.50	4.4767
	mix	26.15	14.90	9.86	7.01	32.14	26.10	58.39	78.28	59.38	86.92	39.91	**3.9037**
	AMD^{2}G	28.78	16.92	13.20	8.31	33.49	25.04	57.02	80.91	60.15	87.36	**41.12**	4.0225
GPT-2	target	8.92	4.53	2.70	1.55	21.11	6.46	18.47	75.87	57.68	87.83	28.51	7.7622
	mix	10.74	5.20	3.05	1.73	24.67	10.11	25.49	75.11	56.74	86.77	29.96	7.6871
	AMD^{2}G	13.89	6.90	3.80	2.61	23.42	11.30	32.54	78.89	59.23	87.90	**32.05**	**7.6083**
BART	target	26.52	14.81	10.17	7.50	31.25	29.39	60.04	76.93	58.41	86.75	40.18	3.9784
	mix	25.64	15.23	11.06	8.46	32.73	29.47	61.73	78.32	59.93	86.39	40.90	3.4973
	AMD^{2}G	29.80	16.80	11.02	8.42	35.43	30.91	62.97	78.36	59.38	86.99	**42.01**	**3.4282**
Music													
Transformer	target	32.49	18.20	12.53	9.45	34.04	13.65	30.98	77.17	61.17	86.58	37.63	4.8624
	mix	36.37	21.23	13.98	9.72	39.19	17.12	39.06	81.93	65.84	88.75	41.32	4.2834
	AMD^{2}G	39.66	22.97	14.41	9.51	40.84	19.89	40.77	82.28	66.63	88.22	**42.52**	**4.1936**
CPT	target	33.52	19.00	11.76	7.67	36.78	18.01	43.09	81.77	64.14	89.76	40.55	3.6496
	mix	36.13	20.71	13.42	9.03	38.57	20.97	48.82	81.95	65.10	88.98	42.37	3.3757
	AMD^{2}G	37.41	21.24	14.23	10.05	39.80	21.50	50.68	82.63	65.89	89.64	**43.31**	**3.2297**
GPT-2	target	27.13	14.37	9.40	6.67	32.74	13.42	31.68	79.34	62.54	88.21	36.55	4.0748
	mix	32.83	17.96	11.81	8.27	34.10	14.71	34.45	79.28	62.99	87.21	38.36	3.5407
	AMD^{2}G	35.98	19.49	13.83	9.53	35.00	18.81	44.34	81.99	63.31	88.40	**41.07**	**3.3353**
BART	target	34.97	21.09	14.46	10.69	40.84	22.74	47.80	82.91	66.93	89.59	43.20	3.1210
	mix	36.02	22.35	15.88	12.04	41.94	22.55	49.43	82.21	67.23	88.68	43.83	3.2139
	AMD^{2}G	38.25	23.99	15.81	14.62	43.46	23.53	53.89	82.59	68.09	89.69	**45.39**	**3.0367**
Travel													
Transformer	target	34.63	27.17	22.57	19.43	47.26	11.55	25.24	80.44	70.83	86.92	42.60	2.3491
	mix	36.03	28.52	23.65	20.58	46.85	12.34	31.90	83.90	72.15	90.22	44.61	2.3350
	AMD^{2}G	36.30	28.40	24.63	22.68	46.96	14.16	33.41	81.58	69.39	89.82	**44.73**	**2.2389**
CPT	target	26.80	20.07	15.67	12.84	47.25	19.19	43.12	83.59	72.76	89.25	43.06	1.8730
	mix	35.17	28.41	24.14	21.19	51.19	17.41	40.83	85.24	74.29	90.56	46.84	1.8273
	AMD^{2}G	36.67	28.73	24.41	21.91	51.93	17.07	39.85	86.48	74.96	91.29	**47.33**	**1.8032**
GPT-2	target	32.41	24.07	19.70	16.59	39.80	11.50	28.15	79.60	68.55	87.45	40.78	3.6998
	mix	36.90	28.48	24.03	21.52	39.27	14.57	34.07	80.81	68.73	88.66	43.70	3.1826
	AMD^{2}G	37.65	30.09	25.88	23.18	44.60	12.07	27.44	83.11	71.77	89.41	**44.52**	**3.0171**
BART	target	31.49	23.14	17.45	13.64	47.29	15.10	35.70	84.23	72.90	90.81	43.17	1.6796
	mix	31.25	24.27	19.45	16.27	49.91	20.88	44.68	85.06	74.07	90.80	45.66	1.6497
	AMD^{2}G	36.23	28.93	24.18	20.72	51.57	19.85	41.85	84.75	74.34	91.25	**47.37**	**1.6307**
E-Commerce													
Transformer	target	13.51	7.81	5.15	3.55	20.48	5.42	16.54	62.11	49.47	68.15	25.22	2.3847
	mix	13.87	8.00	5.09	3.51	20.62	9.11	30.43	62.79	49.45	69.53	27.24	3.2139
	AMD^{2}G	14.92	9.71	6.49	5.97	21.84	9.37	31.70	64.67	51.54	69.17	28.54	2.3438
CPT	target	13.25	8.28	5.70	3.97	23.07	12.02	36.63	63.42	51.04	70.80	28.82	2.1533
	mix	15.55	9.67	6.91	5.33	22.29	12.88	40.92	63.83	50.41	71.87	29.97	2.5117
	AMD^{2}G	16.54	8.72	5.66	4.05	24.17	12.69	41.93	64.38	50.97	71.71	30.08	2.1168
GPT-2	target	7.21	3.69	2.17	1.37	13.70	3.01	9.91	60.58	46.49	71.09	21.92	2.2633
	mix	5.52	2.64	1.62	1.16	10.50	3.60	11.01	58.52	44.01	69.00	20.76	2.2301
	AMD^{2}G	7.44	3.13	1.46	0.76	12.50	5.36	18.34	60.49	44.77	71.60	22.58	2.2175
BART	target	14.94	9.04	6.10	4.18	21.95	13.23	38.38	62.88	50.43	70.25	29.14	2.0949
	mix	15.60	10.56	8.23	6.99	22.30	13.78	40.75	62.39	49.92	71.05	30.16	1.9479
	AMD^{2}G	14.04	9.25	6.50	4.70	23.66	18.61	51.59	62.97	50.61	70.91	31.28	1.8750
Medical													
Transformer	target	13.50	8.94	6.30	3.13	42.44	1.95	4.41	80.71	70.83	82.43	31.46	1.9193
	mix	15.72	9.93	6.58	3.27	40.04	6.93	18.02	79.17	69.12	81.27	33.01	2.4432
	AMD^{2}G	18.50	11.33	7.24	3.98	37.81	9.39	25.10	78.39	67.17	81.52	34.04	1.8951
CPT	target	14.03	9.29	6.46	2.49	40.85	13.68	29.23	78.31	68.46	81.43	34.42	1.9884
	mix	19.88	12.50	8.32	4.62	39.37	13.75	36.04	78.69	67.54	82.21	36.29	1.6892
	AMD^{2}G	21.90	13.69	8.48	4.37	39.99	14.34	37.87	79.40	68.18	82.78	37.10	1.6861
GPT-2	target	19.13	11.12	6.82	3.08	35.33	4.09	10.46	80.67	70.22	83.00	32.39	6.0602
	mix	23.70	14.16	8.90	4.33	36.42	4.53	12.10	81.14	70.92	83.32	33.95	5.7444
	AMD^{2}G	23.94	15.02	8.30	4.17	36.67	4.78	14.60	82.45	71.45	83.23	34.46	5.7559
BART	target	13.05	8.84	6.35	3.01	42.69	8.22	15.05	79.97	70.55	82.21	32.99	1.9705
	mix	16.67	11.12	7.79	4.43	42.77	11.72	27.45	79.74	69.65	82.79	35.41	1.9147
	AMD^{2}G	22.97	12.29	7.94	4.56	41.22	12.29	29.75	80.97	68.91	83.04	36.39	1.7770

Table 5.3: Overview of results. **Bold** indicates the best result. **target** represents the result of training on the corresponding domain training set. **mix** represents the result corresponding to the mixed training set of all domains. **AMD^{2}G** represents the result based on the AMD^{2}G framework. **AVE** represents the average performance. **PPL** refers to perplexity.

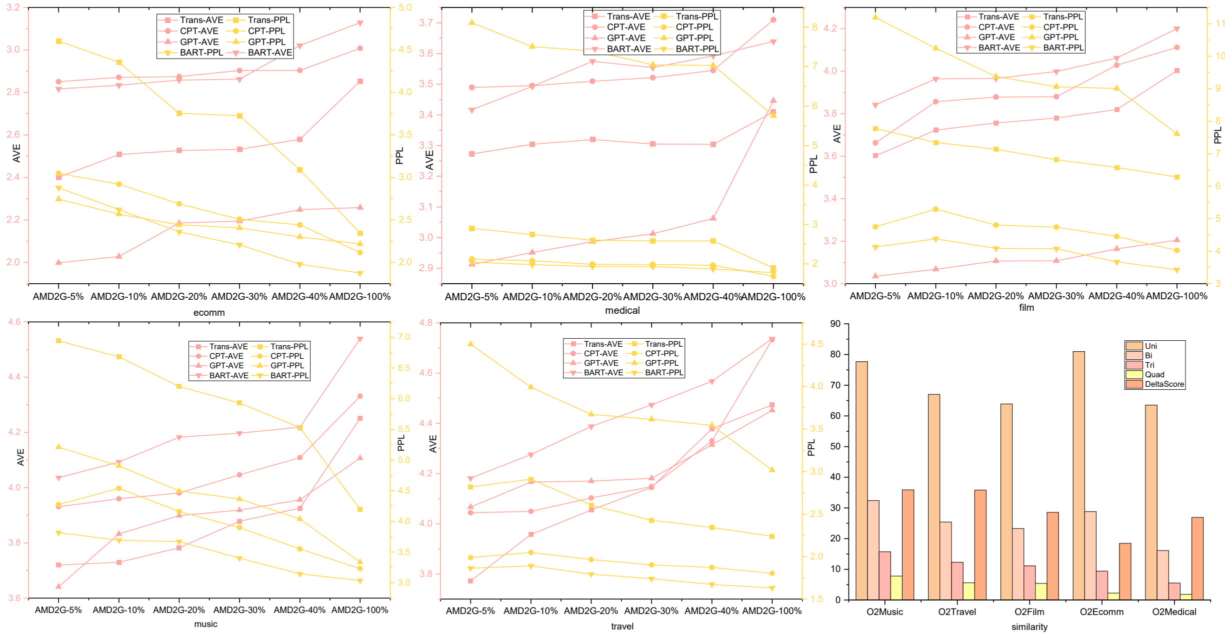


Figure 5.3: The first 5 pictures show the trend of average performance and the trend of PPL as the training data changes in five domains. The last one is n-gram similarity score (i.e., Uni, Bi, Tri, and Quad) and average performance gain trend (i.e., DeltaScore) of models based on **AMD^{2}G** compared to direct training on the target domain corpus. To highlight the trend, we multiply the DeltaScore value by 1000.

5.1.5 Results and Analysis

Overall Results Table 5.3 reports the experimental results of AMD^{2}G in five domains. Compared with training directly on the target domain training set, AMD^{2}G demonstrates absolute advantages in five domains. Specifically, the average performance of the four models has improved by 1.85% in the e-commerce domain, 2.69% in the medical domain, 2.86% in the film domain, 3.59% in the music domain, and 3.58% in the travel domain, compared with training directly in the target domain. Figure 5.3 shows that the performance of some models using 30% target domain training corpus has achieved competitive performance based on the AMD^{2}G framework compared to training on the target domain training set. When using 40% target domain training corpus, the performance based on AMD^{2}G is equivalent to training directly on the target domain training set. These results fully demonstrate the effectiveness of AMD^{2}G.

According to Table 5.2, even though the two domains are quite different, they still share some expression paradigms, which accounts for why the AMD^{2}G framework is effective. The first stage of the AMD^{2}G framework allows the model to learn shared paradigms between different domains, and the second stage allows the model to learn domain-specific features, which can maximize the learning of domain-agnostic features and reduce the mutual influence between domain features. Compared with the performance of training when mixing all domain corpora, the performance of models based on AMD^{2}G exhibits absolute advantages, which confirms that features between domains can interfere with each other and has a negative impact on model per-

Model	Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Rouge-L	Dist-1	Dist-2	Embed A/E/G			AVE↑	PPL↓
						Film							
BART	TS-NET	27.75	14.43	9.76	6.55	33.94	28.45	61.44	78.22	58.42	85.77	40.47	5.6721
	AMD^{2}G	29.80	16.80	11.02	8.42	35.43	30.91	62.97	78.36	59.38	86.99	**42.01**	**3.4282**
GRU	DA-NET	15.55	11.01	7.38	6.33	22.44	11.07	23.08	67.43	54.76	83.55	30.26	25.6935
	AMD^{2}G	16.70	13.52	8.43	8.01	25.38	12.61	25.78	69.35	56.05	84.42	**32.02**	**21.3866**
						Travel							
BART	TS-NET	37.22	20.99	15.22	12.68	40.87	22.45	51.09	80.67	66.53	87.02	43.47	5.0060
	AMD^{2}G	38.25	23.99	15.81	14.62	43.46	23.53	53.89	82.59	68.09	89.69	**45.39**	**3.0367**
GRU	DA-NET	17.86	14.55	10.22	8.55	27.69	14.58	27.82	71.84	56.62	85.33	33.51	12.6600
	AMD^{2}G	19.22	15.62	12.64	10.33	27.71	16.20	28.44	73.05	57.22	85.44	**34.59**	**10.3700**
						E-Commerce							
BART	TS-NET	13.69	10.04	6.72	4.55	21.79	16.93	29.02	60.44	49.79	69.94	28.29	1.9044
	AMD^{2}G	14.04	9.25	6.50	4.70	23.66	18.61	51.59	62.97	50.61	70.91	**31.28**	**1.8750**
GRU	DA-NET	9.33	5.66	3.78	2.28	16.64	4.57	15.77	60.54	48.72	66.12	23.34	7.7743
	AMD^{2}G	11.62	6.44	4.44	3.22	20.09	5.67	17.66	60.62	49.90	67.03	**24.67**	**7.1090**

Table 5.4: The performance of **AMD^{2}G** compared with other baselines in film, travel, and e-commerce domains.

formance. We will discuss in detail the impact of domain similarity on model performance in the next section. Table 5.4 reports the performance comparison of AMD^{2}G and other baselines. We can observe that AMD^{2}G has certain performance advantages compared to TS-NET and DA-NET. TS-NET uses a distillation mechanism to transfer domain knowledge, which relies on a large amount of target domain data. Low resource settings will severely impact distillation results. DA-NET utilizes a dynamic attention mechanism for multi-domain feature fusion. While it preserves domain-specific knowledge, it tends to overlook domain-independent features, thereby hindering its ability to effectively utilize features from other domains. The AMD^{2}G adopts a two-stage strategy to retain domain-agnostic and domain-specific features, and domain-independent features can be adapted to the target domain and achieve data enhancement effects.

Impact of Domain Similarity Domain similarity is a key impact factor for model performance. We perform further experiments to analyze the impact of similarity between domains on model performance. The last one in Figure 5.3 shows the distribution of domain similarity based on n-gram and the average performance gain of models based on AMD^{2}G. We find that the similarity based on 1-gram does not accurately reflect the similarity relationship between domains because it is greatly affected by the unified placeholder. In fact, scores based on n-grams with $n > 2$ better reflect the similarity of domains, because the common expression paradigm is based on more than two words. According to the 4-gram similarity score, the data enhancement effect based on the AMD^{2}G is more obvious for domains with high similarity scores to a certain extent. The more similar the domains are, the more helpful the knowledge provided by other domains will be to the target domain. An exception is the medical domain. Although the similarity score based on 4-gram is low, models based on the AMD^{2}G framework have achieved a certain degree of gain. The domain characteristics of the medical domain are relatively obvious. After de-domaining, the expression paradigms can be aligned to a certain extent. This can be derived from the similarity scores based on n-grams. This is why the AMD^{2}G framework can work in the medical domain.

Impact of Dataset Size In order to explore the impact of data set size, we conduct experiments on 5%, 10%, 20%, 30%, 40%, and 100% of the training set, respectively. Note that 100% of the training set refers to the total amount of data under low resources. Figure 5.3 reports the changing trends of model performance with dataset size in five domains. We can observe that the average performance (i.e., AVE) basically increases gradually as the data size increases, and the PPL decreases as the data set size increases. Models are more sensitive to the size of the data set under low resource conditions. Even a small increase in the data set will make the performance of models increase. The increase in the data set will allow models to learn more different examples, making models cover more test examples.

Model	Corpus	E-Commerce			Film			Travel		
		Fluency	Relevance	Diversity	Fluency	Relevance	Diversity	Fluency	Relevance	Diversity
BART	target	0.377	0.833	0.062	0.579	0.875	0.083	1.034	0.920	0.116
	mix	0.464	1.025	0.085	0.662	0.965	0.098	1.142	0.946	0.104
	AMD^{2}G	**0.522**	**1.150**	**0.113**	**0.784**	**1.110**	**0.115**	**1.206**	**1.012**	**0.133**
GPT-2	target	0.311	0.753	0.065	0.466	0.782	0.076	0.972	0.938	0.096
	mix	0.472	0.975	0.086	0.673	0.950	0.120	1.133	0.955	0.110
	AMD^{2}G	**0.514**	**1.040**	**0.082**	**0.762**	**1.132**	**0.112**	**1.174**	**1.102**	**0.128**

Table 5.5: The results of the human evaluation in e-commerce, film, and travel domains.

Model	Method	E-Commerce			Film			Travel		
		Fluency	Relevance	Diversity	Fluency	Relevance	Diversity	Fluency	Relevance	Diversity
BART	TS-NET	0.456	0.814	0.065	0.677	1.067	0.108	1.110	0.864	0.105
	AMD^{2}G	**0.522**	**1.150**	**0.113**	**0.784**	**1.110**	**0.115**	**1.206**	**1.012**	**0.133**
GRU	DA-NET	0.237	0.755	0.052	0.508	0.833	**0.117**	0.456	0.867	**0.088**
	AMD^{2}G	**0.307**	**0.774**	**0.069**	**0.553**	**0.912**	0.099	**0.542**	**0.955**	0.086

Table 5.6: Comparison with the human evaluation of baselines in e-commerce, film, and travel domains.

Human Evaluation Results We conduct the human evaluation of BART and GPT-2 on three domains to further confirm the effectiveness of AMD^{2}G. To evaluate the consistency of the results assessed by annotators, we employ Pearson's correlation coefficient (Sedgwick, 2012). This coefficient is 0.25 on diversity, 0.68 on relevance, and 0.77 on fluency, with $p < 0.0001$ and below 0.001, which demonstrates high correlation and agreement. The results of the human evaluation are shown in Table 5.5. Compared to training directly in the target domain and on mixed corpora, models based on the AMD^{2}G enjoy a significant advantage in relevance and diversity. Specifically, models based on AMD^{2}G enjoy an average advantage of 20.40% in fluency, 24.1% in relevance, and 3.1% in diversity in three domains compared with models trained directly in the target domain. Compared with models trained on mixed corpora, models based on AMD^{2}G enjoy an average advantage of 6.9% in fluency, 10.2% in relevance, and 1.3% in diversity in three domains. The experimental results show the effectiveness of the AMD^{2}G framework. AMD^{2}G can effectively reduce the mutual interference of domain features and strengthen the learning of domain-agnostic features. Table 5.6 reports the performance comparison of AMD^{2}G and other baselines. The human results show that the AMD^{2}G framework still outperforms the TS-NET and DA-NET. Specifically, the model's average performance, based on AMD^{2}G, is 9.8% higher than TS-NET and 4.3% higher than DA-NET.

5.1.6 Sum-Up

We propose a simple and effective data augmentation framework, AMD^{2}G, for multi-domain low-resource dialogue generation. The domain characteristics of the corpus can be removed through domain dictionaries constructed by LLMs. Models trained on a domain-independent corpus can reduce the interference of different domain features when models learn domain-independent features. Domain adaptation training can adapt the learned domain-independent features to the target domain. Experiments on four models in five domains demonstrate the effectiveness of the AMD^{2}G framework. Compared with other baselines, the AMD^{2}G framework has obvious advantages. AMD^{2}G provides an alternative solution for low-resource multi-domain dialogue generation.

5.2 GNN-Based Parameter-Efficient Fine-Tuning Inspired by Information Flow

This section corresponds to the following work:

> Shuzhou Yuan, **Ercong Nie**, Michael Färber, Helmut Schmid, and Hinrich Schuetze. 2024. GNNavi: Navigating the Information Flow in Large Language Models by Graph Neural Network. In Findings of the Association for Computational Linguistics: ACL 2024, pages 3987–4001, Bangkok, Thailand. Association for Computational Linguistics.

Declaration of Co-Authorship. I conceived the idea of drawing inspiration from the information flow theory for understanding the mechanism of in-context learning paradigms and utilizing the Graph Neural Network (GNN) to explicitly facilitate the information flow process in the fine-tuning. Based on this, Shuzhou Yuan designed the GNN-based parameter-efficient fine-tuning method, dubbed GNNAVI . Shuzhou Yuan implemented GNNAVI with GPT2-XL and Llama2 models and experimented on a series of language understanding tasks. I ran several baseline experiments. Shuzhou Yuan and I worked on drafting the manuscript together. Michael Färber, Helmut Schmid, and Hinrich Schütze are supervisors and provided much valuable advice and feedback.

Summary of This Section

Large Language Models (LLMs) exhibit strong In-Context Learning (ICL) capabilities when prompts with demonstrations are used. However, fine-tuning still remains crucial to further enhance their adaptability. Prompt-based fine-tuning proves to be an effective fine-tuning method in low-data scenarios, but high demands on computing resources limit its practicality. We address this issue by introducing a prompt-based *parameter-efficient fine-tuning (PEFT)* approach. **GNNAVI** leverages insights into ICL's information flow dynamics, which indicate that label words act in prompts as anchors for information propagation. GNNAVI employs a *Graph Neural Network (GNN)* layer to precisely guide the aggregation and distribution of information flow during the processing of prompts by hardwiring the desired information flow into the GNN. Our experiments on text classification tasks with GPT-2 and Llama2 show that GNNAVI surpasses standard prompt-based fine-tuning methods in few-shot settings by updating just 0.2% to 0.5% of parameters. We compare GNNAVI with prevalent PEFT approaches, such as prefix tuning, LoRA, and Adapter in terms of performance and efficiency. Our analysis reveals that GNNAVI enhances information flow and ensures a clear aggregation process.

5.2.1 Motivation

Large language models (LLMs) show remarkable In-Context-Learning (ICL) capabilities by learning from prompts with demonstrations (Wan et al., 2023; Sun et al., 2023; Patel et al., 2023; Mekala et al., 2023; Ko et al., 2023), with the exponential growth in model sizes. However, fine-tuning LLMs still remains essential for further enhancing their adaptability (Zhang et al., 2023b). Prompt-based fine-tuning (Schick and Schütze, 2021a; Ma et al., 2024), adopting objectives that simulate the language modeling process, emerges as a viable technique, particularly in low-data settings (Gao et al., 2021). Yet, the substantial computational demands of Full-Parameter Fine-Tuning (FPFT), which updates billions of parameters, pose a practical challenge. In fact, optimizing a relatively small subset of an LLM's parameters can significantly improve its performance (Ding et al., 2023), paving the way for Parameter-Efficient Fine-Tuning (PEFT) methods. These methods include Adapter (Houlsby et al., 2019), Prompt-Tuning (Lester et al., 2021), Prefix Tuning (Li and Liang, 2021), and LoRA (Hu et al., 2022). They offer alternatives to FPFT, but are often not tailored to the prompt-based fine-tuning of LLMs.

Recent advances in understanding the ICL mechanism offer a new avenue for PEFT of LLMs. ICL's success in leveraging few-shot demonstrations and prompts (Brown et al., 2020) has motivated the adoption of prompt-based fine-tuning for moderately sized language models in a few-shot learning manner (Ma et al., 2023a; Schick and Schütze, 2021c). Recognizing the specific features of fine-tuning LLMs within the framework of ICL, we propose **GNNAVI** , a novel PEFT method designed expressly for prompt-based learning. Our method draws inspiration from recent insights into the underlying process of ICL from an information flow perspective, particularly the role of label words in the prompt (Wang et al., 2023c). Label words act as anchors with two functions: aggregating information from context words and directing this information to the last token for accurate predictions. GNNAVI incorporates this understanding through the integration of a Graph Neural Network (GNN) layer (Kipf and Welling, 2017; Hamilton et al., 2017)

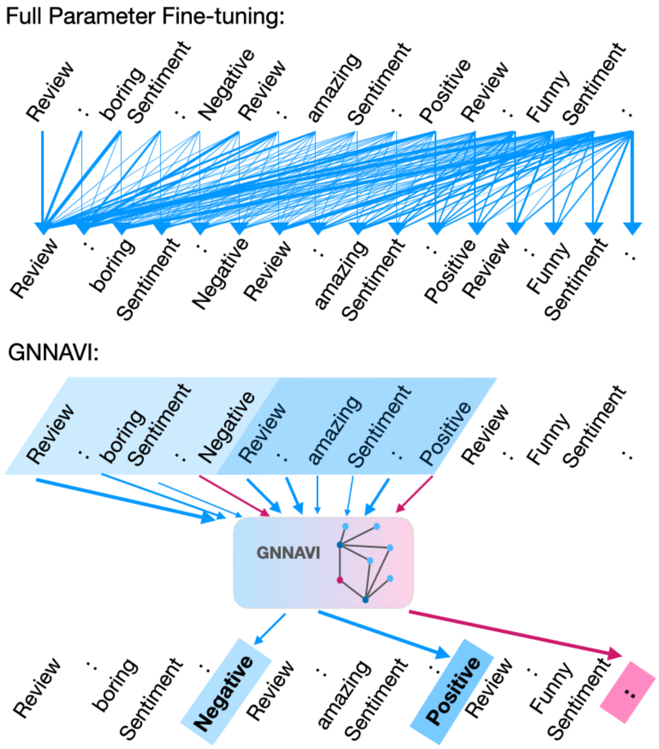


Figure 5.4: Visualization of Full Parameter Fine-tuning (FPFT) and GNNAvi from the perspective of information flow (top words to bottom words). Without GNNAvi, tokens interact with every preceding word in FPFT, leading to confusion in information flow. Conversely, in GNNAvi, label words aggregate information from preceding words (blue path), and the final token aggregates information from the label words (pink path), resulting in a clearer information aggregation process.

into LLMs, optimizing the prompt-based fine-tuning process by navigating the information flow within prompts, as visualized in Figure 5.4. Following the paths of information flow, we insert a GNN layer into the deep layers[6] of the LLM. We treat the input text as a graph, where each token serves as a node, and connect these nodes according to the paths of information flow.

As a PEFT method, GNNAvi adopts a lightweight fine-tuning strategy, updating only the parameters of the GNN layer. Experimenting with few-shot training examples on GPT2-XL (Radford et al., 2019) and Llama2 (Touvron et al., 2023a), GNNAvi achieves remarkable results with just 0.2% of the trainable parameters of the full model, consistently outperforming FPFT and other PEFT methods across various classification tasks. Additionally, we analyze the attention interaction between tokens and find that GNNAvi demonstrates a more stable and clear information aggregation process compared to FPFT.

In summary, our contributions are:

[6] We use "deep layers" to refer to the last few layers of the LLM. For instance, in GPT2-XL, there are 48 layers, with the last 12 layers considered as deep layers in our work.

1. We propose a novel PEFT method, GNNAVI, inspired by the information flow perspective of LLMs. GNNAVI effectively navigates the information aggregation process in LLMs.

2. We apply GNNAVI to text classification tasks with few-shot training examples, outperforming baselines while updating only 0.2% to 0.5% of parameters.

3. Our work sheds light on the application of GNNs in NLP and provides novel insights for future research. To the best of our knowledge, we are the first to utilize GNNs to enhance the performance of LLMs from the information flow perspective.

5.2.2 Background

Parameter-Efficient Fine-Tuning (PEFT) PEFT focuses on enhancing language model performance on downstream tasks by optimizing a small number of parameters, instead of fine-tuning all parameters (Ding et al., 2023). Various PEFT strategies have been explored. Addition-based methods only train modules or parameters added to the model, such as Adapter (Houlsby et al., 2019), Prompt tuning (Lester et al., 2021), and Prefix tuning (Li and Liang, 2021). Specification-based methods selectively fine-tune specific parameters in the original model while keeping the remainder frozen, such as BitFiT (Ben Zaken et al., 2022). Reparameterization-based methods transform existing parameters into a more parameter-efficient form, such as LoRA (Hu et al., 2022). Recent advancements in PEFT research have increasingly prioritized memory efficiency, aiming to enable the training of LLMs with minimal computational resources, such as MeZO (Malladi et al., 2023) and HiFT (Liu et al., 2024). Our proposed PEFT method is designed specifically for LLMs and draws upon the intricacies of how LLMs process and learn from prompts.

GNN for NLP GNNs are predominantly utilized in NLP tasks involving structural input, such as graph-to-text generation (Gardent et al., 2017) and graph-enhanced question answering (Zhang et al., 2022b). Previous approaches employ GNNs to encode complex graph and node representations. For instance, Koncel-Kedziorski et al. (2019) introduced Graph Transformer, which extends graph attention networks (Veličković et al., 2018) for encoding scientific graph inputs, while Li et al. (2021) utilize GNNs to encode knowledge graphs and align them with text embeddings from pretrained language models. Additionally, GNNs serve as auxiliary tools for pretrained language models to encode complex structural information for AMR-to-text generation (Ribeiro et al., 2021). Unlike prior work, we leverage GNNs for information aggregation based on the perspective of information flow.

5.2.3 Methodology

In this subsection, we elaborate on the details of our approach. We begin by detailing the GNNAVI architecture, followed by the task formulation.

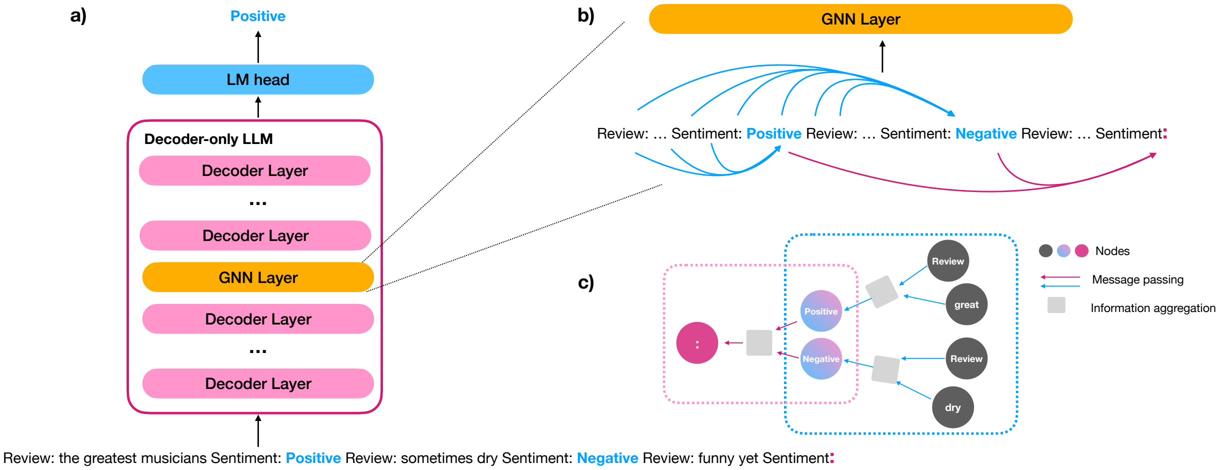

a)
Positive
LM head
Decoder-only LLM
Decoder Layer
...
Decoder Layer
GNN Layer
Decoder Layer
...
Decoder Layer
Review: the greatest musicians Sentiment: Positive Review: sometimes dry Sentiment: Negative Review: funny yet Sentiment:
b)
GNN Layer
Review: ... Sentiment: Positive Review: ... Sentiment: Negative Review: ... Sentiment:
c)
Review
great
Positive
Negative
Review
dry
:
Nodes
Message passing
Information aggregation

Figure 5.5: Visualization of GNNAVI with an example of sentiment analysis, where label words and the last token are highlighted in blue and pink, respectively. a) The GNN layer is integrated into a decoder-only LLM. The LLM processes a prompt containing demonstrations and generates the next token as the prediction. b) The input text is transformed into a graph, with tokens as nodes and information flow paths as edges. c) Visualizing the working mechanism of the GNN: Node representations are updated by aggregating information from incoming nodes. To maintain simplicity, not all nodes are listed.

5.2.3.1 Architecture of GNNAVI

Intuition Wang et al. (2023c) demonstrated that the working mechanism of LLM follows specific paths of information flow. The label words in the input prompt serve two roles for the final predictions: acting as information aggregators by gathering information from their preceding words and propagating the aggregated information to the last token position where the prediction is generated. Building upon their insights, we posit that navigating the flow of information aggregation can enhance both the efficiency and effectiveness of LLMs. Leveraging the GNN's proficiency in information aggregation at the graph level, we explore LLMs from a graph theory perspective and utilize GNN as a tool to guide the information flow.

Working Mechanism We illustrate the working mechanism of GNNAVI in Figure 5.5. For example, in a sentiment analysis task, the prompt comprises one demonstration from each class and the text to be classified. An LLM processes this prompt layer by layer. The GNN layer is inserted after the l-th decoder layer of the LLM[7]. Receiving the token representations from the l-th layer, the GNN layer learns node representations by aggregating information from incoming nodes. Subsequently, the node representations are propagated to the next layer in LLM as hidden

[7] In our preliminary experiments, GNNAVI performs optimally when the GNN layer is inserted in the last quarter of the layers in LLM. Thus, we add the GNN layer after the 42nd layer of GPT2-XL and after the 28th layer of Llama2-7b in our experiments. A detailed analysis is conducted in §5.2.6.

states. The nodes are connected following the paths of information flow. As depicted in Figure 5.5(b), the label words '*Positive*' and '*Negative*' aggregate information from their preceding tokens and pass the information to the last token '*:*' of the prompt. In case the label word is tokenized into subtokens, we use the first subtoken to serve as the label word, following previous work (Zhao et al., 2021; Wang et al., 2023c). We freeze the pretrained parameters of the LLM during training and update only the parameters in the GNN layer.

Graph Neural Network The graph neural network aggregates information from incoming nodes to model graph and node representations by message passing. To formulate an NLP task on a graph level, we consider the input text as a graph. We define a directed graph $\mathcal{G}$ as a triple $(\mathcal{V}, \mathcal{E}, \mathcal{R})$ with a set of nodes $\mathcal{V} = \{v_1, \ldots, v_n\}$ (one node for each token), a set of relation types $\mathcal{R}$[8], and a set of edges $\mathcal{E}$ of the form (v, r, v') with $v, v' \in \mathcal{V}$, and $r \in \mathcal{R}$. Each node v_i is associated with a feature vector x_i, which is the token representation of the i-th token in the l-th layer. In Figure 5.5, for instance, the first token '*Review*' is connected with the label token '*Positive*'. This edge is represented by the triple $(\textit{Review}, aggregate, \textit{Positive})$, where $aggregate$ denotes an edge directed towards a label node.

The node representations in the GNN layer are updated by aggregating the information from incoming nodes. The aggregation algorithms vary across different GNN architectures. For example, the learning process of Graph Convolutional Network (GCN) (Kipf and Welling, 2017) is formulated as:

$$h_v = \sigma\left(W \sum_{v' \in N(v)} \frac{h_{v'}^{(l)}}{|N(v)|}\right) \tag{5.2}$$

where h_v denotes the updated node representation of v, $h_{v'}^{(l)}$ denotes the token representation of its neighbouring nodes from l-th decoder layer, σ is the activation function, W is the trainable parameter of GNN, $N(v)$ includes all the neighbouring nodes of v.

We also include another GNN architecture, GraphSAGE (Hamilton et al., 2017), in our studies, which involves a more complex learning process:

$$h_v = \sigma\left(W\left(h_v^{(l)} \oplus \text{AGG}(\{h_{v'}^{(l)}, \forall v' \in N(v)\})\right)\right) \tag{5.3}$$

The concatenation function $\oplus$ concatenates aggregated information with the node's current representation, and the aggregation function AGG compiles message passing from incoming nodes using techniques such as mean, pool, and LSTM.[9] We visualize the information aggregation process of GNN in Figure 5.5(c).

5.2.3.2 Task Formulation

In our work, we implement prompt-based fine-tuning for text classification tasks. Our goal is to predict the correct class given a few examples. We reformulate the task as a language modeling problem. Let M be a language model with vocabulary V, and let $\mathcal{L}$ be a set of label words. The

[8] In our work, we only consider one relation type: the directed edge from node v to node v'.

[9] We apply mean aggregation to GraphSAGE in this work.

training set $\mathcal{T}$ consists of pairs (s, l), where s is a sequence of tokens from the vocabulary V and l is a label word from the set $\mathcal{L}$. In a sentiment analysis task, for instance, we define a pattern $\mathcal{P}(s, l)$ which associates a text $s =$'Nice performance' and a label word $l =$'Positive' as follows:

Review: Nice performance. Sentiment: Positive

For a k-class classification task, we sample one demonstration per class from the training set $\mathcal{T}$, and concatenate them with the text s to be classified to form the prompt $X(s)$:

$$X(s) = \mathcal{P}(s_1, l_1) \oplus \ldots \oplus \mathcal{P}(s_k, l_k) \oplus \mathcal{P}(s, \varepsilon) \tag{5.4}$$

$\oplus$ denotes the concatenation of the input demonstrations and ε is the empty string. A more intuitive example is shown in Figure 5.5. The language model reads the prompt $X(s)$ and predicts the next token l, which is the label assigned to s. M is initialized with pretrained parameters ϕ, and fine-tuned by minimizing the cross-entropy loss:

$$\ell = - \sum_{(s,l) \in \mathcal{T}} \log p_\phi(l | X(s)) \tag{5.5}$$

$p_\phi(., .)$ returns the probability which M assigns to the correct label l. In our work, we randomly select one demonstration per class to form the prompt and remove it from $\mathcal{T}$. The training examples are then sampled from the remaining samples in $\mathcal{T}$.

5.2.4 Experiments

Datasets We implement text classification tasks using five commonly used datasets from different domains, including **SST-2:** Stanford Sentiment Treebank Binary for sentiment analysis (Socher et al., 2013); **EmoC:** EmoContext for 4-label emotion classification (Chatterjee et al., 2019); **TREC:** Text REtrieval Conference Question Classification (TREC) for question type classification containing 6 types (Li and Roth, 2002; Hovy et al., 2001); **Amazon:** binary classification for Amazon reviews (McAuley and Leskovec, 2013); **AGNews:** AG's news topic classification dataset for topic classification with 4 labels (Zhang et al., 2015).

Models As GNNavi is built on the basis of decoder-only LLMs, we select two large language models, both with over 1 billion parameters, and equip them with GNNavi . Specifically, we choose GPT2-XL with 1.6 billion parameters (Radford et al., 2019) and Llama2 with 7 billion parameters (Touvron et al., 2023a). For the GNN layer, we opt for GCN and GraphSAGE, denoted as **GNNavi-GCN** and **GNNavi-SAGE** in the experiments. To integrate GNNavi with GPT2-XL and Llama2, we modify their source codes from Huggingface (Wolf et al., 2020) and utilize GNN models provided by PyTorch Geometric (Fey and Lenssen, 2019).

Baselines We adopt the following baselines for the experiments:

ICL one-shot per class: In-context learning (ICL) follows the scenario where the LLM is initialized with pre-trained parameters and instructed by demonstrations to perform text classification tasks. None of the model parameters are updated. We sample one demonstration per class to form the prompt. The demonstrations used to form the prompt are consistent with those used for other methods under the same random seed.

ICL few-shot per class: To compare with the low-data fine-tuning setting, we implement ICL with 5 additional shots per class as the demonstrations. This setting is comparable to a training set with a size of 5 samples per class. Due to the limited input length of GPT2-XL, AGNews and Amazon are set to 4 additional shots per class.

Low-Rank Adaptation (LoRA): LoRA is a PEFT method that reduces the number of trainable parameters by injecting trainable rank decomposition matrices into each layer of the LLM (Hu et al., 2022). We implement LoRA using the Python library PEFT (Mangrulkar et al., 2022).

Prefix-tuning (Prefix): Prefix-tuning utilizes a soft-prompt strategy, incorporating virtual tokens into the LLM and updating only the parameters of the virtual tokens (Li and Liang, 2021). We implement prefix-tuning using the PEFT library (Mangrulkar et al., 2022). The number of virtual tokens is set to maintain a comparable size of trainable parameters as for GNNAVI .

Adapter: We insert a standard adapter module after the feed-forward sub-layer of each layer in the LLM (Houlsby et al., 2019). The adapter module is added using AdapterHub (Pfeiffer et al., 2020a; Poth et al., 2023).

Full Parameter Fine-tuning (FPFT): Full parameter fine-tuning is implemented as a strong baseline, where all the model parameters are updated during the training process.

Experimental Setting The prompt is designed following the template in Equation 5.4. We take one demonstration per class to form the prompt and append the sample to be predicted at the end of the prompt. The templates for the prompt are presented in Table 5.7. $[S]$ denotes the demonstration selected to form the prompt, $[L]$ represents the label word of the demonstration, and $[S_i]$ denotes the sample to be predicted.

Task	**Template**	**Label Words**
SST-2	Review:$[S]$ Sentiment: $[L]$ Review:$[S_i]$ Sentiment:	Positive, Negative
EmoC	Dialogue: $[S]$ Emotion: $[L]$ Dialogue:$[S_i]$ Emotion:	Happy, Sad, Angry, Others
TREC	Question: $[S]$ Answer Type: $[L]$ Question: $[S_i]$ Answer Type:	Abbreviation, Entity, Description, Person, Location, Number
Amazon	Review: $[S]$ Sentiment: $[L]$ Review: $[S_i]$ Sentiment:	Positive, Negative
AGNews	Article: $[S]$ Answer: $[L]$ Article: $[S_i]$ Answer:	World, Sports, Business, Technology

Table 5.7: Template for prompt.

Following a few-shot learning setting, we experiment with different numbers of training samples, namely 5, 10, 20, 50, 100, and 200 samples per class. The training samples are randomly selected from the original training set. Another 1000 samples from the original training set are sampled as the validation set, and 1000 samples from the original test set are used for evaluation.[10] The accuracy on the validation set is employed to identify the best-performing model, which is subsequently evaluated on the test set. We report the average accuracy over five random seeds. We present the hyperparameters for GNNavi and other baselines in Table 5.8. The models were trained using NVIDIA A100-SXM4-40GB GPUs. Due to limited resources, the batch size was set to 1, and full parameter fine-tuning of Llama2 was implemented using 8 bits. We observed that for Llama2, GNNavi and other PEFT methods were sensitive to the selection of prompts with very few training samples, and thus could not achieve optimal performance. To address this, we replaced these results by using another random seed to change the demonstrations in the prompt.

Hyperparameter	**GNNavi**	**Prefix**	**Adapter**	**LoRA**	**FPFT**
learning rate	1e-2	1e-2	5e-5	5e-4	5e-5
optimizer	Adam	Adam	AdamW	AdamW	AdamW
epochs	50	50	50	50	50
early Stop	15	15	15	15	15
random seed	[0, 42, 312, 411, 412, 421, 520, 1218]				
virtual tokens	-	40(GPT2), 150(Llama2)	-		

Table 5.8: Hyperparameters for GNNavi and baselines.

5.2.5 Results

We report the results with 5 and 200 training examples in Table 5.9, which reflect the performance under the scenarios where only limited training examples are available and sufficient training examples are provided, respectively. Full results are presented in Appendix F.

Overall Performance Observing the results of GPT2-XL, GNNavi remarkably rivals ICL, FPFT, and other parameter-efficient baselines. Under the low-data setting of 5 training examples, both GNNavi-GCN and GNNavi-SAGE outperform FPFT by over 13%, achieving higher accuracy than other PEFT methods by 0.4% to 21%. Increasing the number of training examples to 200, the average performance of GNNavi improves to 89.64% and outperforms other baselines.

Similar to GPT2-XL, GNNavi achieves the best performance with Llama2 among all the baselines. With only 5 training examples, GNNavi-SAGE achieves 2.77% higher average accuracy than FPFT. Compared with other PEFT methods, GNNavi shows higher average accuracy from 1.8% to 35%. And with 200 training examples, GNNavi-GCN achieves 92.24% average accuracy, outperforming FPFT, Prefix-tuning, Adapter, and LoRA.

[10] The original test set of SST-2 contains less than 1000 samples, so we keep the original test set for evaluation.

Method	#Param	SST-2	EmoC	TREC	Amazon	AGNews	Average	#Param	SST-2	EmoC	TREC	Amazon	AGNews	Average
	GPT2-XL							**Llama2**						
$k = 0$														
ICL	-	55.44	6.48	54.68	53.32	72.12	48.41	-	67.55	9.60	70.36	94.98	84.14	65.33
$k = 5$														
ICL	-	63.17	6.30	57.68	53.67	50.43	46.25	-	86.93	20.18	45.72	92.30	80.16	65.06
LoRA	2.5M	91.98	50.60	75.20	88.80	**85.20**	78.36	4.2M	**95.42**	64.20	**88.40**	91.80	86.60	85.28
Prefix	6.1M	59.13	73.46	32.92	60.00	75.40	60.18	39.3M	50.96	58.56	21.36	49.36	25.78	41.20
Adapter	15.4M	79.82	76.00	**79.60**	**91.45**	81.25	81.62	198M	50.92	**84.05**	18.80	49.45	24.80	45.60
FPFT	1.6B	62.13	61.30	65.28	73.00	80.82	68.51	6.7B	94.63	61.92	81.72	**95.86**	**87.58**	84.34
GNNAVI-GCN	2.6M	**84.31**	75.48	76.72	90.90	83.16	**82.11**	16.8M	94.56	78.30	83.2	94.00	86.25	86.63
GNNAVI-SAGE	5.1M	81.95	**78.70**	77.92	88.66	82.88	82.02	33.6M	92.91	80.12	80.80	95.66	86.06	**87.11**
$k = 200$														
LoRA	2.5M	**90.83**	80.80	90.80	82.00	86.20	86.13	4.2M	91.29	**86.80**	93.60	95.80	90.40	91.32
Prefix	6.1M	50.92	80.18	69.80	59.80	79.08	67.96	39.3M	48.35	81.72	45.68	52.28	27.54	51.11
Adapter	15.4M	88.65	80.70	**96.60**	92.30	**89.80**	89.61	198M	50.92	85.05	88.20	49.45	81.50	67.57
FPFT	1.6B	68.97	73.70	80.16	74.82	85.34	76.60	6.7B	**95.64**	79.90	**96.76**	96.12	**91.44**	91.97
GNNAVI-GCN	2.6M	90.67	78.82	91.88	92.94	89.20	88.70	16.8M	95.36	82.85	95.50	**96.45**	91.05	**92.24**
GNNAVI-SAGE	5.1M	90.46	**82.68**	92.32	**93.44**	89.28	**89.64**	33.6M	95.30	81.94	94.76	95.96	90.68	91.73

Table 5.9: Results of different training methods (accuracy). k denotes the number of training examples per class, #Param denotes the number of trainable parameters. The best scores are highlighted with **bold**.

	SST-2	EmoC	TREC	Amazon	Agnews
GPT2-XL	4.7×	6.3×	4.1×	3.9×	3.4×
Llama2	4.3×	2.4×	1.6×	1.4×	1.2×

Table 5.10: The ratio by which the training process is accelerated for one training epoch for GNNAVI-GCN compared to FPFT.

Efficiency Analysis GNNAVI significantly reduces the number of trainable parameters compared to the baselines for both GPT2-XL and Llama2. GNNAVI-GCN for GPT2-XL achieves the highest average accuracy with 5 training examples containing only 2.5 million trainable parameters, which is 615 times smaller than FPFT, six times smaller than Adapter, twice smaller than Prefix, and similar to LoRA. As for Llama2, GNNAVI saves over 6.6 billion trainable parameters compared to FPFT and achieves better results. GNNAVI-GCN also updates fewer parameters than Prefix and Adapter. LoRA contains fewer trainable parameters than GNNAVI-GCN in Llama2, but the performance of LoRA cannot compete with GNNAVI-GCN and GNNAVI-SAGE . Table 5.10 shows that by saving a significant amount of training parameters, GNNAVI-GCN speeds up the training process by a factor of up to 6 compared to FPFT.

Influence of Training Examples Adding more training examples improves the accuracy for GNNAVI and most baselines. As depicted in Figure 5.6, GNNAVI consistently outperforms other methods as the number of training examples increases. While other methods also show improvement with more training examples, the extent of improvement is not as consistent as for GNNAVI, particularly for Prefix and Adapter.

Figure 5.7 shows the performance of GNNAVI for the different tasks as a function of the number of training examples. We observe that the effect of adding training examples is similar

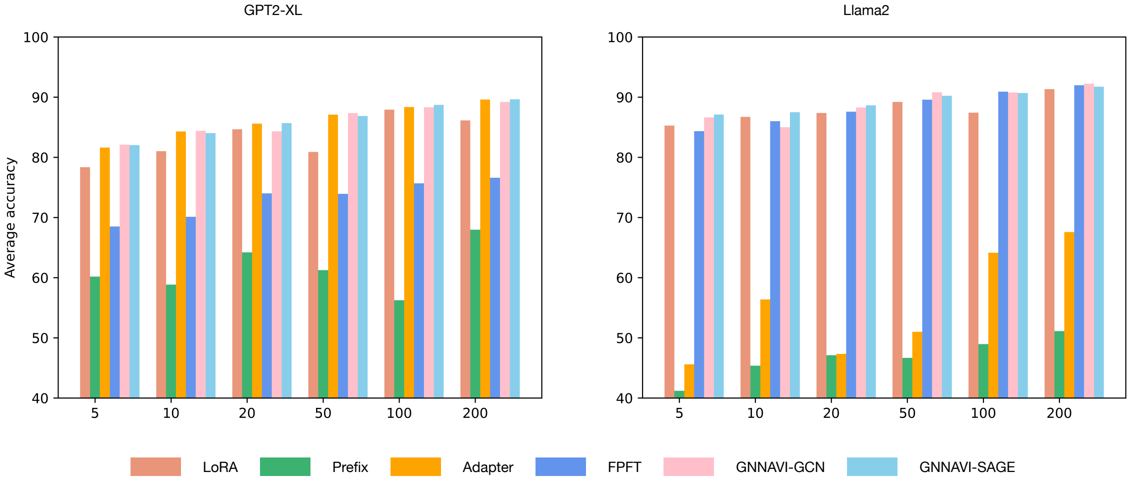


Figure 5.6: Results of average accuracy with different numbers of training examples. The x-axis denotes the number of training examples per class.

for both GPT2-XL and Llama2. Notably, adding more training examples yields significant improvements, especially in low-data settings (e.g. with 10, 20, and 50 training examples) where GNNAVI shows a substantial improvement, except for EmoC. However, the significance diminishes when more than 50 training examples are provided, the improvement is not as pronounced here as in low-data settings.

5.2.6 Ablation Study

In this subsection, we delve into the influence of the position where the GNN layer is inserted in the LLM and investigate the effects of removing one of the information flow paths on performance. All of these studies are conducted using GNNAVI-SAGE with 5 training samples per class under the experimental settings outlined in §5.2.4.

Position of GNN Layer The position where the GNN layer is inserted significantly impacts the model's performance. Figure 5.8 illustrates the performance of GNNAVI when the GNN layer is inserted at different locations in GPT2-XL. With the exception of EmoC, all tasks exhibit lower performance when the GNN layer is added in the first 10 layers of GPT2-XL. Performance improves as the GNN is added in deeper layers, reaching peak accuracy around the 44th layer. Subsequently, accuracy declines until the last layer. This trend may stem from the gradual initiation of the information flow process in the early layers of LLM, where the GNN's influence is limited due to insufficient token interaction. Conversely, in the final layers, the information flow process is nearly complete, rendering it too late for the GNN to guide effectively. Despite variations in performance changes across tasks, the average performance suggests that the optimal placement for the GNN layer is between the 38th and 42nd layers for GPT2-XL.

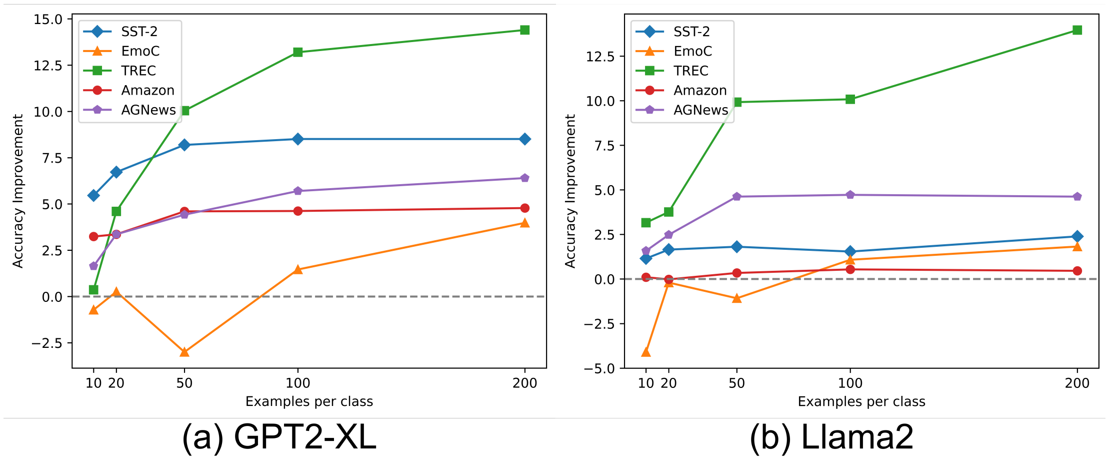


Figure 5.7: The improvement gained by adding training examples for GNNAVI-SAGE , compared to using 5 training examples per class.

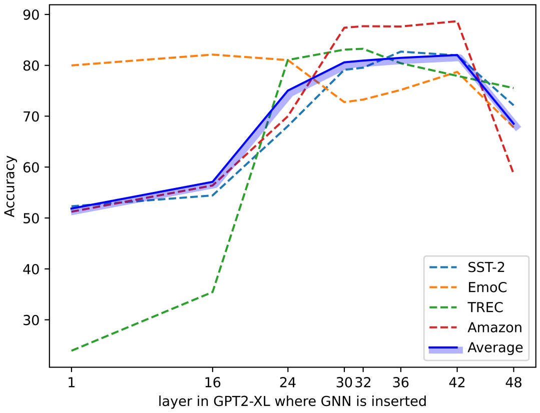


Figure 5.8: Performance Comparison with GNN inserted at various positions in GPT2-XL.

Removal of Information Flow We conduct an ablation study to investigate how removing specific information flow paths affects the results while retaining others. In our approach, we connect the label words to their preceding words to aggregate information and to the last token to distribute the information from the label words. These connections are referred to as the aggregation and distribution paths in the ablation study. As illustrated in Figure 5.9, we remove one path and retain another.

As shown in Table 5.11, both the aggregation and distribution paths contribute significantly to the performance. Removing either of them results in a decrease in the average accuracy across the five tasks. Except for the two binary classification tasks, SST-2 and Amazon, removing the distribution path causes a greater drop in performance. Based on these results, we conclude that the distribution path plays a more significant role in the information flow process, especially for tasks with more than two labels.

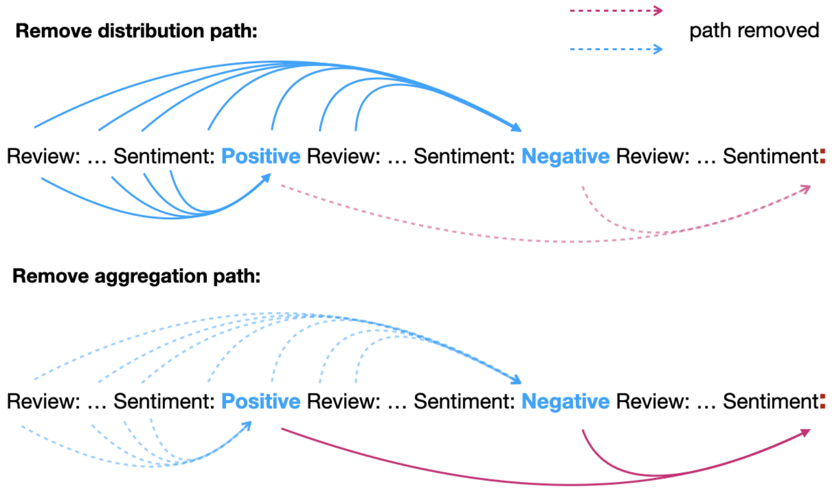

Remove distribution path:
path removed
Review: ... Sentiment: Positive Review: ... Sentiment: Negative Review: ... Sentiment:
Remove aggregation path:
Review: ... Sentiment: Positive Review: ... Sentiment: Negative Review: ... Sentiment:

Figure 5.9: Visualisation of the ablation study on the removal of information flow.

	SST-2	EmoC	TREC	Amazon	Agnews	Average
GNNAVI-SAGE	81.95	78.70	77.92	88.66	82.88	82.02
-aggregation	-0.07	-1.10	-0.68	+0.56	-0.08	-0.27
-distribution	+3.07	-12.88	-2.44	+1.64	-1.44	-2.41

Table 5.11: Ablation Study. Removal of information flow. The name indicates the removed path.

5.2.7 Further Discussion: Information Flow

While the attention mechanism in LLM offers an information flow perspective for interpreting the model's working mechanism (Wang et al., 2023c), it treats the input text as a fully connected graph. In contrast, GNNAVI explicitly connects the context tokens to the label tokens for information aggregation and the label tokens to the final token for information distribution. Thereby, the correct information flow is hardwired into the GNN. There is no need to learn it by adjusting the attention weights. To further investigate the differences in information flow between GNNAVI and FPFT, we utilize the saliency technique (Simonyan et al., 2013) for interpretation. Following the approach of Wang et al. (2023c), we compute the saliency score for each element of the attention matrix using a Taylor expansion (Michel et al., 2019):

$$I_l = \sum_h \left| A_{h,l}^\top \frac{\partial L(x)}{\partial A_{h,l}} \right|, \tag{5.6}$$

where $A_{h,l}$ represents the attention matrix of the h-th attention head in the l-th layer. x is the input, and $L(x)$ is the loss function. The saliency matrix I_l for the l-th layer is obtained by averaging the values across all attention heads. Each element $I_l(i,j)$ of the matrix denotes the significance of the information flow from the j-th word to the i-th word in the prompt.

We employ three quantitative metrics to assess the information flow: S_{agg} measures the information flow of the aggregation path from previous context words to label words, S_{dist} measures the information distribution from label words to the last token, and S_{rest} accounts for other information flow between remaining words excluding S_{agg} and S_{dist}. The average significance of information flow can be formulated as:

$$S = \frac{\sum_{(i,j)\in C} I_l(i,j)}{|C|}, \tag{5.7}$$

where C is the set of all token interactions involved. S_{agg}, S_{dist}, and S_{rest} are calculated as follows:

We utilize $l_1, l_2, \cdots, l_C$ to denote the label word positions, such as 'Positive' and 'Negative', while f represents the final token, such as ':'. Additionally, t denotes other tokens excluding the label and final tokens.

S_{agg} calculates the mean significance of information flow from the previous context words to label words:

$$\begin{gathered} S_{agg} = \frac{\sum_{(i,j)\in C_{tl}} I_l(i,j)}{|C_{tl}|}, \\ C_{tl} = \left\{(l_k, j) : k \in [1, C], j < l_k\right\}. \end{gathered} \tag{5.8}$$

S_{dist} calculates the mean significance of information flow from the label words to the final token:

$$\begin{gathered} S_{dist} = \frac{\sum_{(i,j)\in C_{lf}} I_l(i,j)}{|C_{lf}|}, \\ C_{lf} = \left\{(f, l_k) : k \in [1, C]\right\}. \end{gathered} \tag{5.9}$$

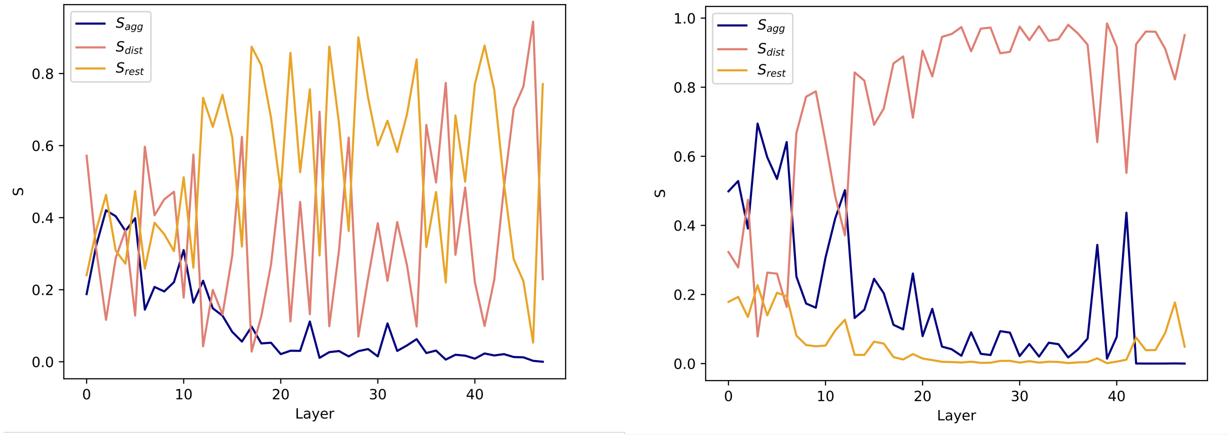
S_{rest} calculates the mean significance of information flow among the rest words, excluding S_{agg} and S_{dist}:

$$\begin{gathered} S_{rest} = \frac{\sum_{(i,j)\in C_{tt}} I_l(i,j)}{|C_{tt}|}, \\ C_{tt} = \{(i,j) : j < i\} - C_{tl} - C_{lf}. \end{gathered} \tag{5.10}$$

As depicted in Figure 5.10, the information flow of GNNAvi appears more stable compared to FPFT. In FPFT, without guided navigation, tokens interact with every preceding word, leading to a trend of confusion between the information flow S_{dist} and S_{rest}. This indicates a struggle to identify the 'right' information for the final prediction. Conversely, GNNAvi adheres to the information flow guided by the GNN, resulting in stable curves that depict a consistent information aggregation process, aligning with the findings of Wang et al. (2023c). Compared to FPFT, the stable curves affirm that GNNAvi serves as a navigator, ensuring the information flows in predefined directions.

5.2.8 Sum-Up

In this section, we propose a novel PEFT method, GNNAvi, leveraging GNN to navigate information flow within LLMs. Specifically tailored for prompt-based fine-tuning, GNNAvi significantly reduces the number of trainable parameters by simply adding a GNN layer into LLMs



(a) FPFT

(b) GNNavi

Figure 5.10: Comparison of information flow between FPFT and GNNAVI for SST-2. Both models are trained with 5 training examples per class.

to guide the information flow within the prompt. GNNAVI outperforms FPFT and other PEFT methods across various classification tasks, even with few training examples. Our work offers insights into handling LLMs from a graph perspective and presents a novel application of GNNs in NLP. Future work could explore different token connectivities for GNNs or utilize GNNs to control the information flow in LLMs.

Chapter 6

Human-Inspired Understanding of Language Models

Summary of This Chapter

A central theme of this dissertation is not only to develop efficient and robust methods for multilingual and low-resource NLP, but also to deepen our understanding of how LLMs process, represent, and sometimes fail at language. As LLMs become increasingly integral to real-world applications, the need for interpretability and human-aligned analysis grows ever more urgent. This chapter addresses this need by exploring human-inspired interpretability methods that probe the internal workings of LLMs, drawing on insights from psycholinguistics, neurolinguistics, and cognitive science. By doing so, it connects the practical advances of earlier chapters with a broader scientific quest: to bridge the gap between surface-level performance and true linguistic competence, and to illuminate the mechanisms underlying both the successes and limitations of modern language models.

Within this framework, the chapter first investigates the internal representations of LLMs through the lens of human cognitive paradigms. We introduce a probing methodology that treats LLMs as both psycholinguistic and neurolinguistic subjects, inspired by experimental traditions in human language research. Specifically, we propose minimal pair probing to disentangle how LLMs encode linguistic form (signifier) and meaning (signified) across languages. This approach enables a fine-grained assessment of the distinction between performance (observable behavior) and competence (underlying knowledge), revealing that LLMs often exhibit stronger mastery of linguistic form than of conceptual meaning (§6.1).

Beyond probing, the chapter advances to neuron-level mechanistic interpretability, focusing on a critical failure mode in multilingual NLP: language confusion in English-centric LLMs. Drawing inspiration from the phenomenon of code-switching in human bilingualism, we employ neuron-level mechanistic analysis to trace how and where language confusion arises within the model's architecture. Using behavioral benchmarks and tools such as TunedLens, we identify confusion points—specific positions in the generation process where unintended language switches occur—and reveal that these are driven by transition failures in the final layers of the

model. Through targeted neuron attribution and editing, we demonstrate that intervening on a small set of critical neurons can substantially mitigate language confusion, achieving results on par with multilingual-aligned models while preserving general competence and output quality. This neuron-level intervention offers a principled and interpretable solution to a persistent challenge in multilingual NLP, and highlights the potential of mechanistic interpretability for building more reliable and human-aligned language technologies (§6.2).

In summary, this chapter integrates human-inspired probing and mechanistic analysis to provide a comprehensive understanding of LLMs' internal representations and failure modes. By bridging cognitive paradigms and model internals, it not only advances the interpretability of language models, but also informs the design of future NLP systems that are both scientifically grounded and practically robust, furthering the monograph's central vision of efficient, inclusive, and human-inspired NLP for all languages.

6.1 Large Language Models as Neuro- vs. Psycholinguistic Subjects

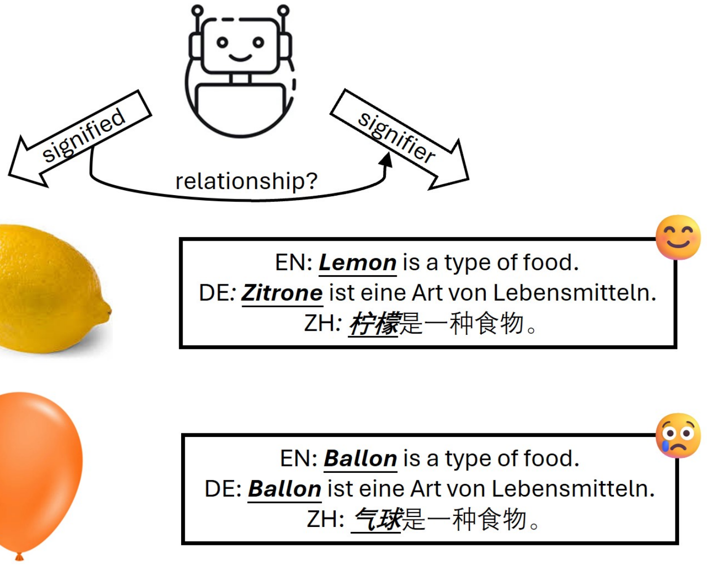
This section corresponds to the following work:

> Linyang He, **Ercong Nie**, Helmut Schmid, Hinrich Schütze, Nima Mesgarani, Jonathan Brennan. 2024. Large Language Models as Neurolinguistic Subjects: Discrepancy between Performance and Competence. In Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria. Association for Computational Linguistics.

Declaration of Co-Authorship. I proposed the research question of using linguistic and conceptual minimal pairs to conduct a comparative investigation of probing and prompting Large Language Models (LLMs). Linyang He framed the research question from the perspectives of treating LLMs as Neurolinguistic and Psycholinguistic subjects. I prepared the minimal datasets for conceptual understanding and ran two baseline experiments (direct probability measurement and metalinguistic prompting). Linyang He implemented the minimal pair probing method and completed the result analysis. Linyange He and I completed the manuscript together. Helmut Schmid, Hinrich Schütze, Nima Mesgarani, and Jonathan Brennan are supervisors of this project and provided advice and guidance.

Summary of This Section

This study investigates the linguistic understanding of LLMs regarding signifier (form) and signified (meaning) by distinguishing two LLM assessment paradigms: psycholinguistic and neurolinguistic. Traditional psycholinguistic evaluations often reflect statistical rules that may not accurately represent LLMs' true linguistic competence. We introduce a neurolinguistic approach, utilizing a novel method that combines minimal pairs and diagnostic probing to analyze activation patterns across model layers. This method allows for a detailed examination of how LLMs represent form and meaning, and whether these representations are consistent across languages. We found: (1) Psycholinguistic and neurolinguistic methods reveal that language performance and competence are distinct; (2) Direct probability measurement may not accurately assess linguistic competence; (3) Instruction tuning won't change much competence but improve performance; (4) LLMs exhibit higher competence and performance in form compared to meaning. Additionally, we introduce new conceptual minimal pair datasets for Chinese (COMPS-ZH) and German (COMPS-DE), complementing existing English datasets.



EN: ***Lemon*** is a type of food.
DE: ***Zitrone*** ist eine Art von Lebensmitteln.
ZH: ***柠檬***是一种食物。

EN: ***Ballon*** is a type of food.
DE: ***Ballon*** ist eine Art von Lebensmitteln.
ZH: ***气球***是一种食物。

Figure 6.1: Illustration of LLMs processing the same signified across different signifiers.

6.1.1 Background and Motivation

Large Language Models (LLMs) have demonstrated remarkable reasoning, linguistic, arithmetic, and other cognitive abilities. The advent of LLMs has reignited cross-disciplinary discussions about what sorts of behavior are "intelligence", even if the intelligence exhibited by LLMs may differ from human intelligence (Sejnowski, 2023). LLMs have drawn the attention of researchers from various fields, including linguistics, cognitive science, computer science, and neuroscience, who investigate how LLMs develop and exhibit these capabilities.

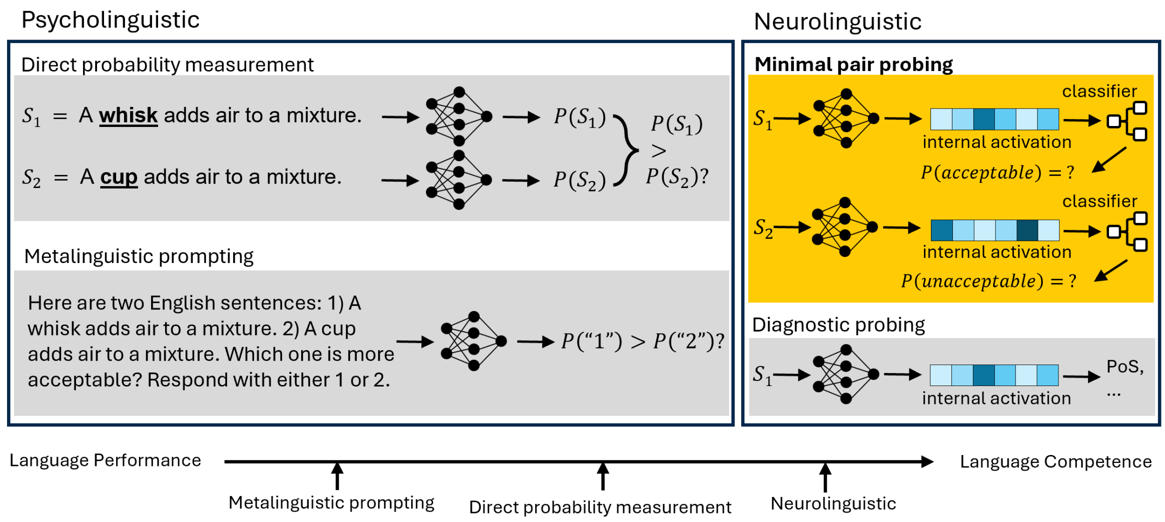


Figure 6.2: Psycholinguistic vs. Neurolinguistic Paradigm. Both direct probability measurement and metalinguistic prompting can be considered as psycholinguistic methods, while minimal pair probing (He et al., 2024a) and other diagnostic probing are neurolinguistic.

There is currently a heated debate about whether LLMs understand human language or whether their performance is simply the product of complex statistical relationships (Mitchell and Krakauer, 2023). A central aspect of this debate concerns the nature of LLMs' linguistic representations. Using the semiotic framework of language proposed by De Saussure (1989), which distinguishes between the signifier (form) and the signified (meaning), we can inquire into the extent to which LLMs comprehend the form and meaning, and how form and meaning intertwist with each other. Is LLMs' understanding of language meaning merely a statistical outcome based on their grasp of language form? When different languages express a shared concept with distinct forms, do LLMs create similar representations for these variations? How can we better understand the representations of form and meaning in these systems that support the observed patterns of performance?

The underlying processes remain unclear due to the opaque nature of neural networks. Therefore, we need appropriate methods to assess their true linguistic understanding. Drawing inspiration from the cognitive study on human language processing, we propose that the assessment of LLMs can be divided into two primary paradigms: As illustrated in Figure 6.2, the psycholinguistic paradigm measures the model's output probabilities, directly reflecting the model's behavior and performance. The neurolinguistic paradigm delves into the internal representations of LLMs.

When treating LLMs as psycholinguistic subjects, their responses may leverage their grasp of form, relying on statistical correlations, to create an illusion of understanding meaning. This enables LLMs to produce structurally coherent but not necessarily semantically accurate responses, as their “understanding” is shaped by patterns rather than true conceptual processing (Harnad, 1990; Bender and Koller, 2020; Nie et al., 2024). Consequently, psycholinguistic evaluations tend to reflect performance rather than competence, as they assess external outputs that may not fully capture the underlying linguistic knowledge encoded within the model. This mismatch suggests that psycholinguistic evaluation results might not accurately represent the true linguistic competence of LLMs.

In contrast, examining LLMs as neurolinguistic subjects focuses on internal representations, providing a more direct assessment of competence by moving beyond surface-level biases (Firestone, 2020). To achieve this, we adapted the decoding probing method by He et al. (2024a), referred to as “minimal pair probing”, to analyze how LLMs encode form and meaning across layers. This approach allows for a finer distinction between performance and competence, revealing insights that psycholinguistic methods might overlook.

In order to address questions about whether LLMs maintain consistent underlying representations of the same concept when the form changes across multiple languages, we also create a multilingual minimal pair dataset (COMPS-ZH for Chinese and COMPS-DE for German).

By evaluating LLMs in both psycholinguistic and neurolinguistic paradigms, we found: 1) Psycholinguistic and neurolinguistic results reveal very different patterns, suggesting both paradigms are necessary for a comprehensive understanding of LLMs. 2) Though more intrinsic than metalinguistic prompting, direct probability measurement may still not accurately assess linguistic competence, as it remains influenced by statistical patterns. 3) LLMs acquire competence in linguistic form more easily, earlier, and with greater accuracy than in meaning. 4) As linguistic form varies across languages, LLMs’ understanding of the same concept shifts accordingly, with meaning competence linearly correlated to form. This suggests that the signifier and signified in LLMs may not be independent, and maintaining conceptual representations likely depends on statistical correlations with form.

6.1.2 Psycholinguistic vs. Neurolinguistic Paradigm

6.1.2.1 Cognitive Science Background

Psycholinguistics and neurolinguistics offer distinct yet complementary perspectives on human language processing. Psycholinguistics focuses on the psychological and cognitive processes that enable humans to understand and use language (Field, 2004; Traxler and Gernsbacher, 2011). In contrast, neurolinguistics explores the underlying neural mechanisms and brain structures involved in language processing (Friederici, 2011; Brennan, 2022; Kemmerer, 2022). Both paradigms offer a valuable model for probing the linguistic capacities and potential intelligence of LLMs.

6.1.2.2 In LLM Assessment Research

Psycholinguistic paradigm: direct probability measurement and metalinguistic prompting Recent studies often use prompting to evaluate the linguistic capabilities of LLMs. These implicit tests were referred to as *metalinguistic judgments* by Hu and Levy (2023). However, it is important to note that the performance of LLMs in specific linguistic prompting tasks only indirectly reflects their internal linguistic representations due to the inherent limitations of such prompting tasks: an LLM chat system might give a “reasonable” response just because of the statistical relationships between prompt and reply (Hofstadter, 1995). Hu and Levy (2023) argue that it is uncertain whether the LLMs' responses to metalinguistic prompting align with the underlying internal representations.

Computing a model's probability of generating two minimally different sentences is one way to address these concerns (Hu and Levy, 2023). The minimal difference between the two sentences (e.g., replacement of a single word) makes one sentence acceptable while the other is not (Linzen et al., 2016). Here are two examples for testing grammatical and conceptual understanding, respectively:

(1) *Simple agreement* (Warstadt et al., 2020):

a. The cats annoy Tim. (*acceptable*)
b. *The cats annoys Tim. (*unacceptable*)

(2) *Concept understanding* (Misra et al., 2023):

a. A whisk adds air to a mixture. (*acceptable*)
b. *A cup adds air to a mixture. (*unacceptable*)

A language model is considered to perform correctly on this task if it assigns a higher probability to the acceptable sentence compared to the unacceptable one (Marvin and Linzen, 2018). Researchers have created syntactic, semantic/conceptual, and discourse inference tasks for the minimal pair method. They provide more precise insights into the abilities of LLMs compared to metalinguistic prompting (Futrell et al., 2019; Gauthier et al., 2020; Hu et al., 2020a; Warstadt et al., 2020; Beyer et al., 2021; Misra et al., 2023; Kauf et al., 2023).

Through either metalinguistic judgement or direct probability measurement methods, these tasks essentially treat LLMs as *psycholinguistic* subjects (Futrell et al., 2019). This research paradigm resembles cognitive psychology by having LLMs perform tasks, such as cloze and question answering, and then evaluating their performance without examining the internal representations, in a manner similar to how subjects participate in psychological experiments. Information about the inner workings of a model is inferred either from its output or from the probabilities it assigns to different possible outputs. The internal states of the LLM (i.e. its intermediate layers) are not examined.

Neurolinguistic paradigm: diagnostic probing Another line of research focuses on studying the internal representations, emphasizing a *neurolinguistic* approach to understanding LLMs. Essentially, diagnostic probing methods in evaluating language models can be considered as neurolinguistic paradigms as they examine the internal states of LMs (Belinkov and Glass, 2019; Belinkov, 2022), while the term '*neurolinguistic*' hasn't been applied to the field before. Diagnostic probing involves training a classifier to predict linguistic properties from the hidden states of LMs. Following this paradigm, researchers decode syntactic, semantic, morphological, and other linguistic properties from the hidden states of LMs (Köhn, 2015; Gupta et al., 2015; Shi et al., 2016; Tenney et al., 2019; Hewitt and Manning, 2019; Manning et al., 2020).

6.1.3 Minimal Pair Probing = Minimal Pair + Diagnostic Probing

While prior neurolinguistic approaches have explored internal representations, they often employed coarse-grained datasets and primarily focused on decoding linguistic labels from embeddings, providing a general perspective on the linguistic features encoded in LMs. In contrast, the minimal pair probing method presented by He et al. (2024a) integrates minimal pair design with diagnostic probing. This combination leverages the granularity of minimal pair design and the layer-wise insights of diagnostic probing, thereby enabling a more detailed analysis of internal patterns for form and meaning. We adopt minimal pair decoding as the neurolinguistic paradigm in our work.

Specifically, given an LLM $f : x_{0,1,\ldots,i} \rightarrow x_{i+1}$ trained on dataset $\mathcal{D}_O$, we can extract the hidden state representations $f_l(S)$ of the l-th layer of stimuli S. Given a minimal pair dataset $\mathcal{D}_m$ = $\{(S^i_+, S^i_-), (z^i_+, z^i_-)\}$ with each sentence S having a label z, we have internal representations $f_l(S^i_+)$ and $f_l(S^i_-)$ for each sentence pair. A minimal probing classifier $g : f_l(S) \rightarrow \hat{z}$ is trained and evaluated on $\mathcal{D}_m$, with grammatical/conceptual performance measure $\mathrm{Perf}(f, \mathcal{D}_O, g, \mathcal{D}_m)$.

Note that our focus is on evaluating the linguistic competence of the LLM f itself, i.e., $\mathrm{Perf}(f, \mathcal{D}_O)$, rather than the capacity of the probing classifier g. As suggested by Hewitt and Liang (2019), even untrained or random representations can yield surprisingly high probing accuracy, raising concerns that the classifier may exploit dataset artifacts rather than meaningful representations. To control for the potential bias introduced by g, we construct a random embedding baseline.

Specifically, for each sentence in the dataset, we assign a fixed random vector r, sampled from a Gaussian distribution with the same mean and standard deviation as the real model embeddings $f_l(S)$. Importantly, each sentence is consistently assigned the same random vector across occurrences, preserving instance-level identity that the probing classifier might exploit. This allows us to assess the extent to which task performance can be driven by superficial sentence-level cues rather than meaningful representations. We then compute $\mathrm{Perf}(g, \mathcal{D}_m)$ by training g on these random embeddings, which reflects the inherent predictability or "shortcut" potential of the probing task. Therefore, our performance score incorporates a correction factor based on this random baseline, defined as:

$$\mathrm{Perf}(f, \mathcal{D}_O) \triangleq \mathrm{Perf}(f, \mathcal{D}_O, g, \mathcal{D}_m) \cdot (1 + \frac{0.5 - \mathrm{Perf}(g, \mathcal{D}_m)}{0.5}) \tag{6.1}$$

This formula applies a correction term, penalizing cases where the probing classifier performs well even on random embeddings. When $\mathrm{Perf}(g, \mathcal{D}_m) = 0.5$, the correction factor is 1; if the performance is higher, the factor shrinks toward 0, discouraging overfitting or trivial tasks; if it drops below 0.5, the factor exceeds 1, slightly amplifying the model's score. This ensures that only meaningful representations in f contribute to the final evaluation.

6.1.4 Experiment Setup

6.1.4.1 Datasets

We use minimal pair probing for English, Chinese, and German to assess grammaticality (form) and conceptuality (meaning). Table 6.1 presents the overall dataset information used in our experiments.

Minimal Pair	Duality	Language	# of Pair	Description
BLiMP	Form	English	67, 000	67 tasks across 12 grammatical phenomena
CLiMP	Form	Chinese	16, 000	16 tasks across 9 grammatical phenomena
DistilLingEval	Form	German	8, 000	8 German grammatical phenomena
COMPS	Meaning	English	49, 340	4 types of conceptual relationship
COMPS-ZH	Meaning	Chinese	49, 340	4 types of conceptual relationship
COMPS-DE	Meaning	German	49, 340	4 types of conceptual relationship

Table 6.1: Overview of datasets in our study.

BLiMP BLiMP (Warstadt et al., 2020) is a comprehensive English dataset of grammatical minimal pairs. It consists of minimal pairs for 13 higher-level linguistic phenomena in the English language, further divided into 67 distinct realizations, called paradigms. Each paradigm comprises 1,000 individual minimal pairs, resulting in a total corpus size of 67,000 data points.

CLiMP CLiMP (Xiang et al., 2021) is a corpus of Chinese grammatical minimal pairs consisting of 16 datasets, each containing 1,000 sentence pairs. CLiMP covers 9 major Chinese language phenomena in total, less than the BLiMP dataset due to the less inflectional nature of Mandarin Chinese. The vocabulary of the CLiMP dataset is based on the translation of the BLiMP dataset, with words and features specific to Chinese added.

DistilLingEval DistilLingEval (Vamvas and Sennrich, 2021) is a dataset of German grammatical minimal pairs. It consists of minimal pairs for eight German linguistic phenomena. This dataset contains 82,711 data samples in total.

COMPS COMPS (Misra et al., 2023) is an English dataset of conceptual minimal pairs for testing an LLM's knowledge of everyday concepts (e.g., *a* ***beaver/**** *gorilla has a flat tail*). This dataset contains 49,340 sentence pairs, constructed using 521 concepts and 3,592 properties. Concepts in the pairs constitute 4 types of knowledge relationships: taxonomy, property norms, co-occurrence, and random.

COMPS-ZH and COMPS-DE COMPS-DE and COMPS-ZH are newly developed datasets featuring conceptual minimal pairs in Chinese and German, derived from the English COMPS dataset (Misra et al., 2023). In the realm of multilingual NLP research, it is a common practice to extend English datasets to other languages using human translation, machine translation, or translation assisted by LLMs (Nie et al., 2023a; Wang et al., 2024a; Beniwal et al., 2024).

In this study, to create COMPS-DE and COMPS-ZH from the original English COMPS, we employed a hybrid approach that integrated machine translation with meticulous human verification.

Specifically, we translated the concepts and properties of the English COMPS individually, subsequently merging them to form complete sentences and compose conceptual minimal pairs. The translation process began with the use of the Google Translate API[1], which provided initial translations of concepts and properties into German and Chinese.

Following this, native speakers of Chinese and German manually checked and refined these translations to ensure accuracy and quality. The manual review emphasized two main areas: accuracy of concepts and grammatical consistency of properties. For concepts, the focus was on correcting ambiguities that might arise from machine translation. For properties, attention was given to maintaining grammatical consistency with the original English text, such as ensuring subject-verb agreement, which is particularly challenging in German translations.

In summary, out of 521 concepts, manual corrections were made to 57 entries in the Chinese dataset and 49 in the German dataset. Similarly, out of 3,592 properties, 713 required manual corrections in the Chinese dataset, and 512 in the German dataset. This rigorous process was essential for preserving the integrity and reliability of the translated datasets.

6.1.4.2 Models

In our experiments, we used three open-source LLMs, two English-centric LLMs (Llama2 and Llama3), and one multilingual LLM (Qwen) with a focus on English and Chinese. These models were trained on different amounts of English, Chinese, and German data (see Table 6.2).

Resource Level	**Llama2**	**Llama3**	**Qwen**
English	High	High	High
Chinese	Mid	Mid	High
German	Low	Low	Low

Table 6.2: Resource level for different languages across three LLMs. Note the resource levels are qualitative assessments based on available information, as specific quantitative data is not provided by the developers.

Llama2 and Llama3 Llama2 (Touvron et al., 2023b) and Llama3 (AI, 2024) are two English-centric LLMs that represent an advanced iteration of the Llama foundation models developed by Meta AI (Touvron et al., 2023a). The Llama models were trained on publicly available corpora predominantly in English. Despite this focus, Llama models are also exposed to a limited amount of multilingual data. Llama 1, for example, is pretrained on an extensive scale of corpora comprising over 1.4 trillion tokens, of which less than 4.5% constitute multilingual data from 20 different languages. Llama 2 expands this linguistic diversity, featuring 27 languages, each representing more than 0.005% of the pertaining data. Therefore, English-centric models harness

[1] https://cloud.google.com/translate

multilingual abilities (Lai et al., 2023a). In this work, we use Llama2-7B and Llama3-8B for our experiments.

QWen QWen is a series of LLMs developed by Alibaba Inc. (Bai et al., 2023). Qwen was trained on 2-3 trillion tokens of multilingual pre-training data. It is essentially a multilingual LLM with a focus on English and Chinese. We use the Qwen-7B model in our experiments.

6.1.4.3 Setup for Psycholinguistic Analysis

Direct Direct probability measurement is based on the probability that the model assigns to a sentence. Accuracy is determined by whether the model assigns a higher probability to the grammatically or conceptually correct sentence within the minimal pair.

Meta Metalinguistic prompting involves explicitly asking a question or specifying a task that requires a judgment about a linguistic expression. Following Hu and Levy (2023), we use one prompt for a minimal pair to present both sentences at once. For form tasks, we assign an identifier (1 or 2) to each sentence in the pair, present a multiple-choice question comparing both sentences, and compare the probabilities assigned by the model to each answer option, “1” or “2”. For meaning tasks, we reformulate the property into a question and compare the probabilities of acceptable and unacceptable concepts as sentence continuations. Table 6.3 presents the prompts used in the experiments.

Duality	Method	Example
Form	Direct	{Mice are hurting a waiter, Mice was hurting a waiter}
	Meta	Here are two English sentences: 1) Mice are hurting a waiter. 2) Mice was hurting a waiter. Which sentence is a better English sentence? Respond with either 1 or 2 as your answer. Answer: {1, 2}
Meaning	Direct	{Helmet can absorb shocks, Cap can absorb shocks}
	Meta	What word is most likely to come next in the following sentence (helmet, or cap)? What can absorb shocks? {helmet, cap}

Table 6.3: Prompt examples for baseline methods. The region where we measure probability is marked in color. Correct sentences and answers are in blue; incorrect in red.

6.1.4.4 Setup for Neurolinguistic Analysis

Sentence Embedding We extract the representation of the last token in each sentence from each layer to serve as the representation for the whole sentence. Last token pooling ensures the representation contains the information of all preceding tokens (Meng et al., 2024).

Probing Performance We use logistic regression as the probing classifier and F1 score as the evaluation metric. The score for $\text{Perf}(f, \mathcal{D}_O, g, \mathcal{D}_m)$ and $\text{Perf}(g, \mathcal{D}_m)$ is calculated as the average F1 score across 5 cross-validation folds. Final performance $\text{Perf}(f, \mathcal{D}_O)$ is given by Formula 6.1.

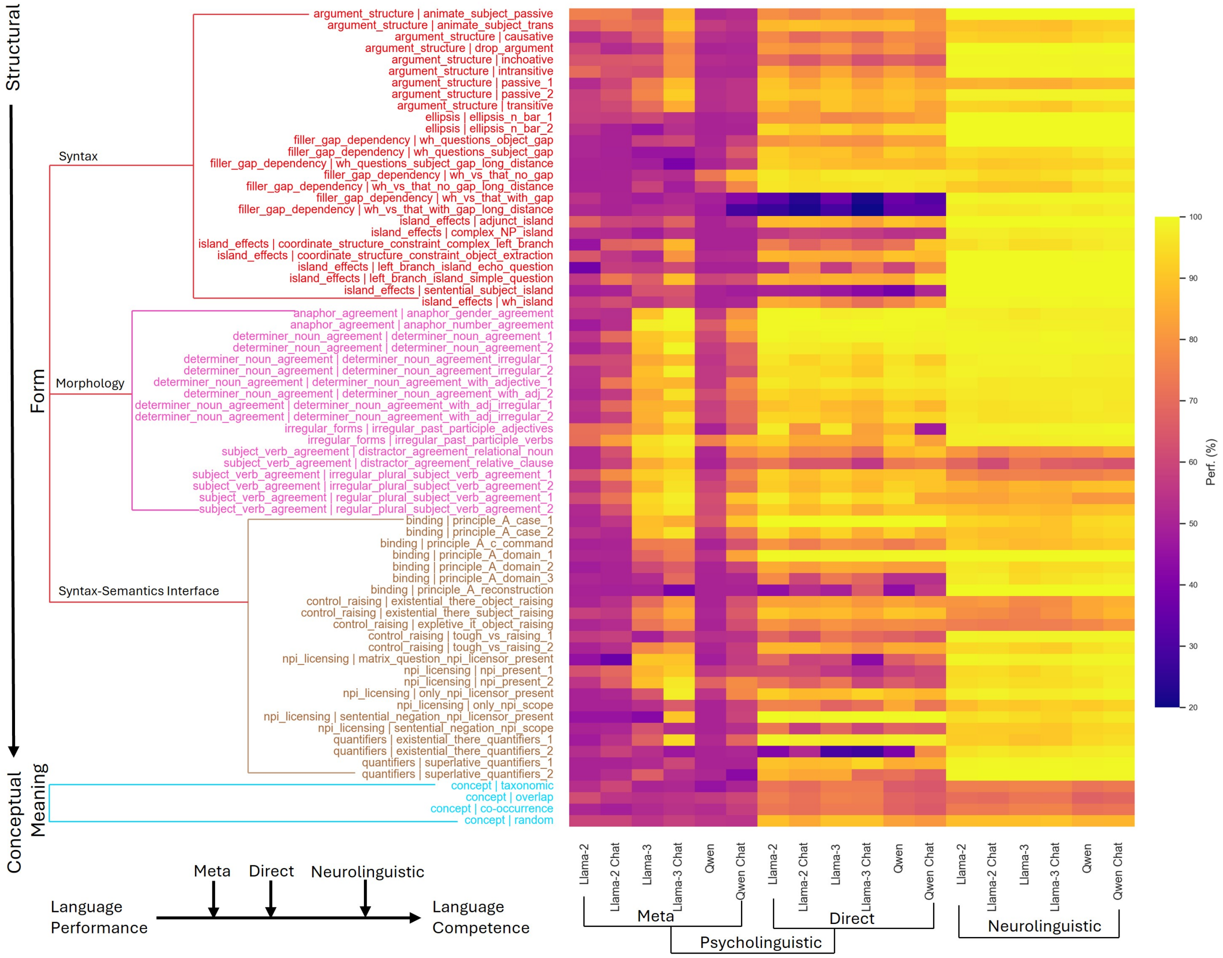


Figure 6.3: Psycholinguistic (meta and direct) and neurolinguistic performance across models and linguistic tasks. The x-axis represents different models and conditions (base and chat), while the y-axis categorizes linguistic tasks based on structural (syntax, morphology, syntax-semantics interface) and conceptual (meaning) levels.

Saturation and Maximum Layer We define the feature learning Saturation Layer as the layer where performance first reaches 95% of the peak on the curve. This layer indicates the number of layers required for the model to adequately learn specific linguistic features, after which its ability to capture these features stabilizes. The Maximum Layer is the layer at which performance reaches its peak.

Unsupervised Analysis We use t-SNE to visualize the sentence embedding of Llama2-7B for English form tasks. We employ PCA to reduce the dimensionality of the sentence embedding to 50 before applying t-SNE.

6.1.5 Results

6.1.5.1 Psycholinguistic vs Neurolinguistic

Figure 6.3 shows the performance of LLMs across all linguistic tasks. Figure 6.4 demonstrates the averaged performance of LLMs across models and 4 levels (syntax, morphology, syntax-semantics interfaces, concept). Figure 6.5 presents the average performance of LLMs across form and meaning tasks for Direct, Meta, and Neuro[2] methods. We use the last layer's performance in the Neuro method when comparing psycho- and neurolinguistic paradigms, as both direct probability measurement and metalinguistic prompting rely on the last layer of LLMs.

Language performance and competence are distinct (Competence > Performance). Figure 6.4 and 6.5 shows distinct results between language performance and competence. Moving from Meta → Direct → Neuro, the evaluation focus gradually shifts from language performance (task execution ability) to language competence (the underlying linguistic ability). Within the same task category, Neuro methods consistently yield higher performance than Direct methods, which in turn outperform Meta methods. This indicates that when evaluating pure linguistic competence, LLMs perform well, but their performance drops when assessed in a task-based setting.

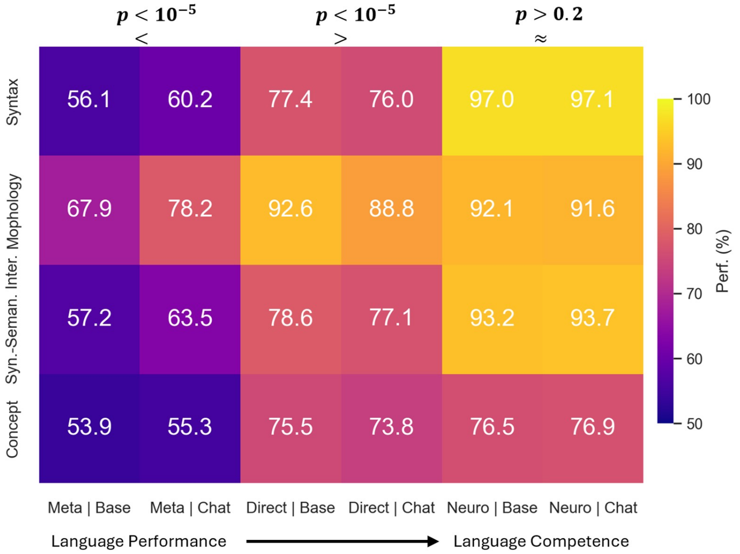


Figure 6.4: Averaged psycholinguistic (meta and direct) and neurolinguistic results across models and tasks. t-tests were conducted on the original (pre-averaging) results between base and chat models, with p-values annotated.

Tasks that emphasize language performance become more difficult, even if their language competence is high. For example, in the Neuro setting, performance on Syntax tasks reaches

[2] We refer to minimal pair probing as Neuro for simplicity.

97%, while in the Meta setting, it drops to 56.1%, showing a significant gap. This suggests that even when an LLM has strong competence in a given task, its performance can significantly decline when assessed under a performance-oriented evaluation.

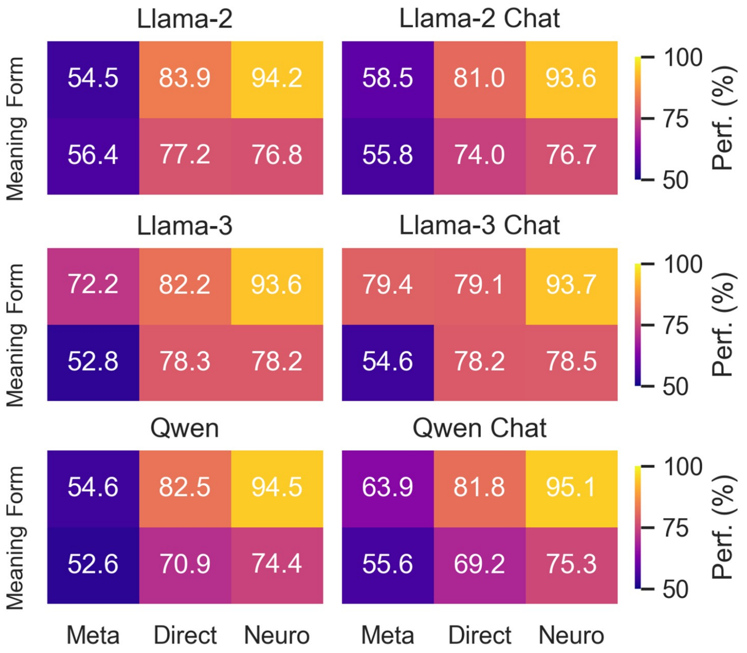


Figure 6.5: Psycholinguistic and neurolinguistic performance for form (morphology, syntax-semantics interface, and syntax) and meaning (concept).

Direct probability measurement might not be the true competence assessment. As the Neuro method measures the internal representations of LLMs directly, it could serve as a reliable ground truth for estimating linguistic competence. Direct probability measurement falls short of achieving this ground truth in form assessment (especially for syntax and syntax-semantics-interface as shown in Figure 6.5).

LLMs exhibit stronger mastery of form than meaning, regardless of performance or competence. As shown in Figure 6.5, LLMs consistently perform better on form-related tasks than on meaning-related tasks. This trend holds regardless of whether the model is a base or chat version. Crucially, this pattern is evident across all evaluation methods. This indicates that LLMs have a stronger grasp of linguistic form than conceptual meaning, whether assessed through task execution or underlying capability.

Instruction tuning won't change much competence but improve performance. Neuro results between the base and chat versions of LLMs reveal that instruction fine-tuning does not significantly alter the language competence of the models (t-test between Neuro-Base vs. Neuro-Chat as shown in Figure 6.4). With instruction fine-tuning (chat versions of LLMs), the Meta

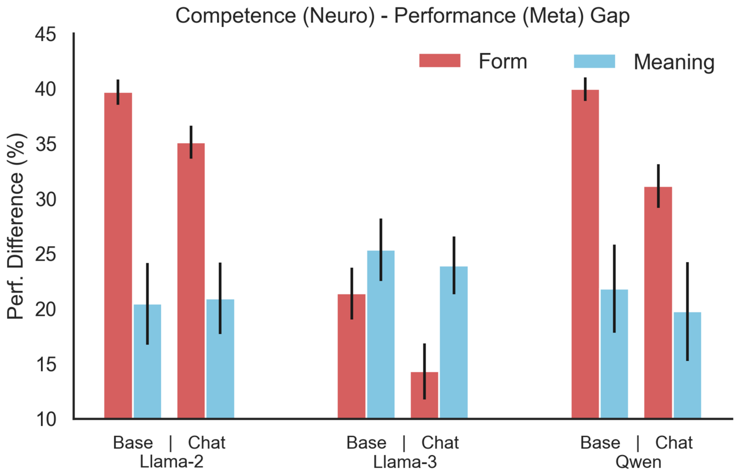


Figure 6.6: Competence and performance gap drops after instruction tuning.

performance on form improves significantly while meaning understanding remains stable. Figure 6.6 illustrates that after instruction tuning, the competence-performance gap (Neuro-Meta) significantly decreases for form-related tasks, while the change for meaning-related tasks remains relatively small. This indicates that fine-tuning with well-designed instructions helps LLMs improve their performance on form-related tasks, bringing them closer to their underlying competence. However, for meaning-related tasks, instruction tuning does not lead to a fundamental improvement in understanding. This indicates that more optimized information access strategies can enhance the external performance of language models, particularly for form-related tasks.

6.1.5.2 Neurolinguistic Analysis

Raw results for English, Chinese, and German can be found in Figure 4, 5, and 6 in Appendix G.

Layer-wise unsupervised dynamics reveal gradual emergence of form features Figure 6.7 illustrates the layer-wise differences between embeddings for grammatically correct and incorrect sentences. In early layers, the embedding difference appears scattered and unstructured, but as depth increases, they form clearer clusters, indicating a progressively refined sensitivity to syntactic correctness. By Layer 16 and beyond, distinct clusters emerge corresponding to syntax, morphology, and syntax-semantics interface. The results demonstrate that LLMs encode grammaticality judgments dynamically across layers, progressively structuring linguistic representations. Moreover, the formation of distinct clusters for different linguistic phenomena in the unsupervised analysis provides supporting evidence for subsequent supervised classification.

Gradual decline in encoding performance from structure to meaning. The results in Figure 6.8-(c) show that the performance scores for conceptual understanding are significantly lower than those for grammatical understanding. This pattern is consistent across all six models, suggesting a universal characteristic of LLMs. Moreover, as illustrated in Figure 6.8-(a),(b),

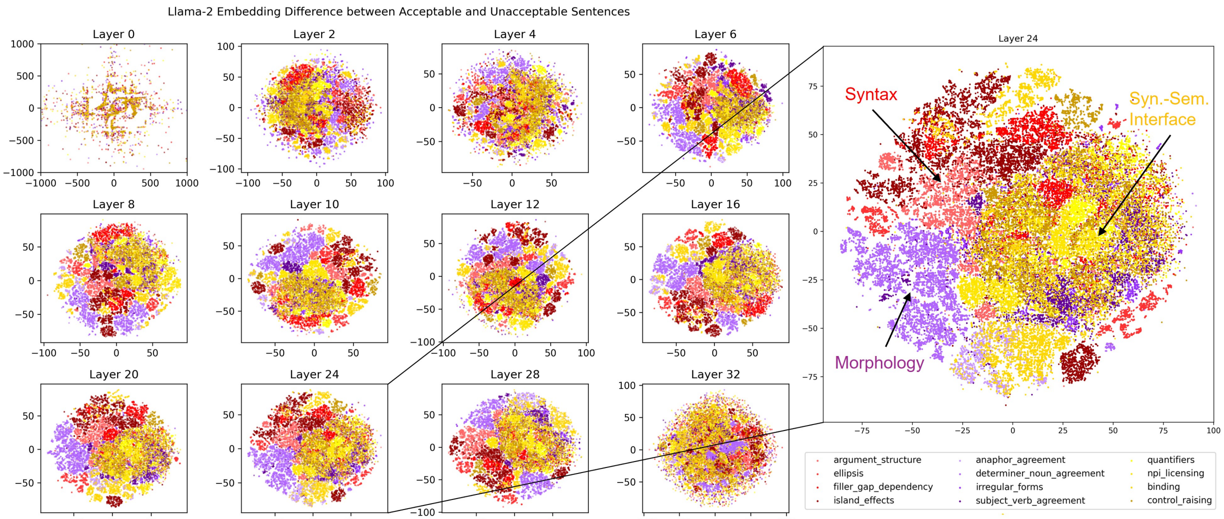


Figure 6.7: t-SNE visualization of embedding differences between acceptable and unacceptable sentences, with red for syntax, purple for morphology, and yellow for the syntax-semantics interface.

the encoding performance progressively declines from more structural tasks to more semantic tasks—spanning syntax, morphology, the syntax-semantic interface, and finally conceptual understanding. This highlights that LLMs encode features less effectively as the tasks shift from structure-focused to meaning-focused.

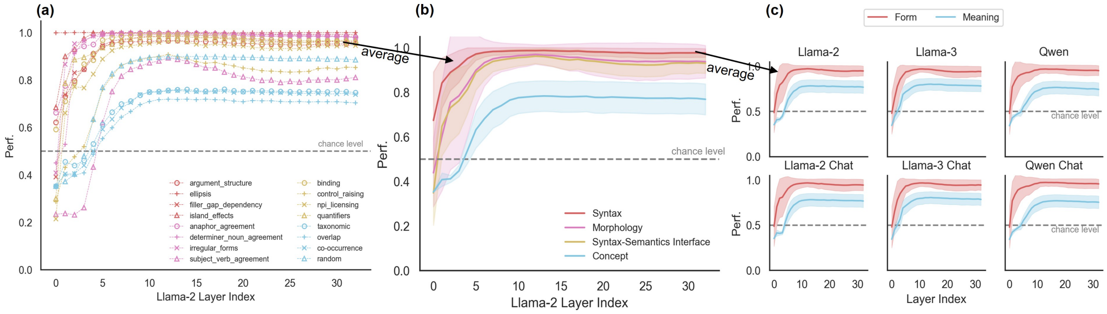


Figure 6.8: **(a)** Neurolinguistic probing performance for 16 tasks in Llama-2, including 4 syntax tasks, 4 morphology tasks, 4 syntax-semantics interface tasks, and 4 conceptual tasks. **(b)** Average probing performance across the four linguistic categories in Llama-2. **(c)** Mean performance comparison between form-related tasks (syntax, morphology, syntax-semantics interface) and meaning-related tasks (concept), aggregated across all six models.

LLMs encode form earlier than meaning. We compute the feature learning saturation and maximum layers for all 12 grammatical tasks and 4 conceptual tasks, averaging them to represent form and meaning, respectively. As shown in Figure 6.9, the saturation and maximum layers for meaning are generally higher than those for form across all six models. This suggests that LLMs stabilize their encoding of grammatical features before conceptual features.

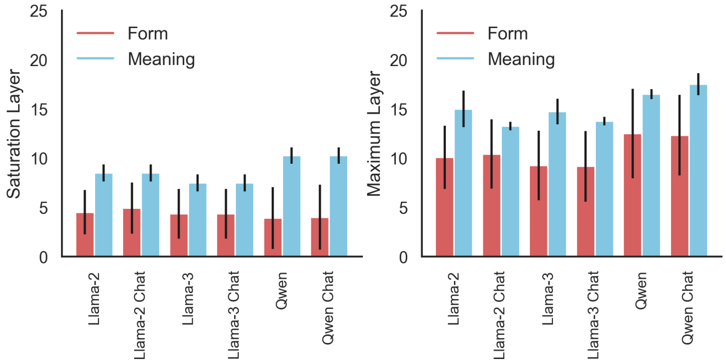


Figure 6.9: Feature learning saturation layer (defined as the first layer reaching 95% of peak performance) and the layer of maximum performance.

Instruction tuning has minimal impact on the internal linguistic representations. As Figure 6.10 shows, performance differences (with and without instruction tuning) for form and meaning remain near zero across all layers, indicating that instruction tuning minimally impacts internal linguistic representations, consistent with our psycholinguistic vs. neurolinguistic analysis.

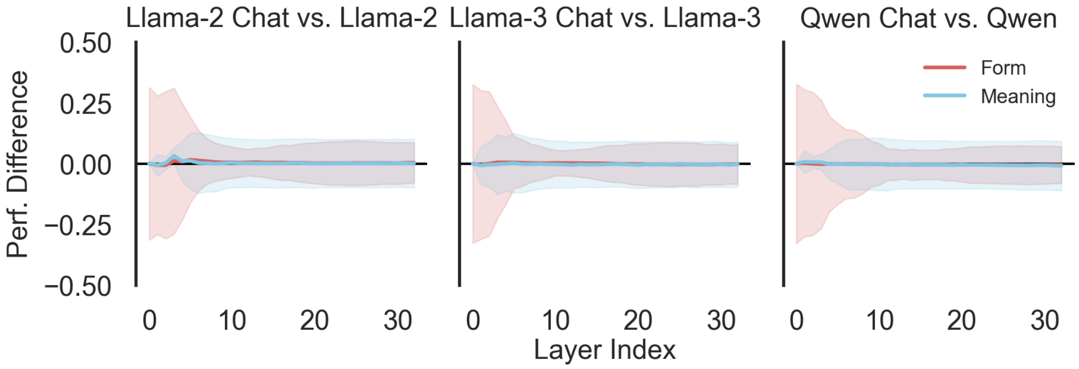


Figure 6.10: Difference in probing performance between base and instruction-tuned models across all layers.

6.1.5.3 Multilingual analysis

How does LLMs' understanding of meaning change when the form (language) varies? Since our COMPS-ZH and COMPS-DE datasets align with the concepts in the English COMPS dataset, we can explore whether LLMs' grasp of different linguistic forms for the same concept correlates with their understanding of meaning across languages. Our previous results suggest that instruction tuning has little influence on the internal representations. Therefore, we focus on the base LLMs here. From Figure 6.11, for all models and languages, form consistently achieves higher performance than meaning, indicating it's easier for LLMs to make a stronger grasp of structural elements compared to conceptual comprehension.

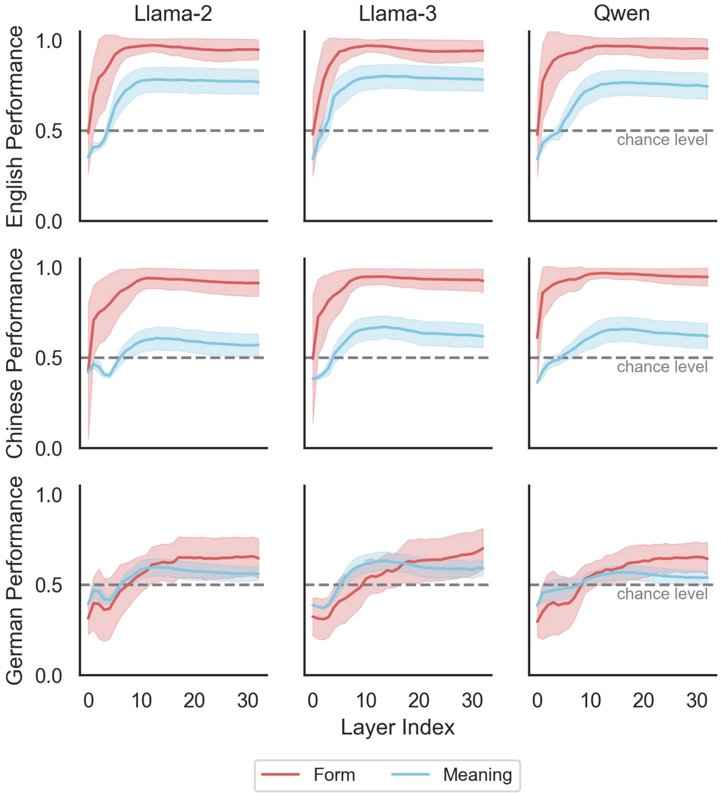


Figure 6.11: Neuro probing results for English, Chinese and German.

6.1.6 Discussion

Language performance vs. competence: probing reveals deeper linguistic understanding than direct probability. Our results demonstrate that neurolinguistic probing uncovers linguistic competencies in LLMs that are not captured by psycholinguistic methods. While Meta performs the worst and Direct performs better, Neuro consistently outperforms both, revealing a systematic underestimation of competence when relying on output-based evaluations.

Hu and Levy (2023) argued that Direct probability measurement, being more intrinsic than metalinguistic prompting, better reflects competence. However, our findings show that even Direct falls short of revealing the full extent of LLMs' linguistic capabilities. Direct relies on the final output layer, which is highly optimized for next-word prediction and thus entangled with task-specific objectives. Prior studies (Hewitt and Manning, 2019; Liu et al., 2019a) have shown that syntactic and general linguistic information is often better represented in intermediate layers than in the final layer. Waldis et al. (2024) also emphasized that output correctness is an insufficient indicator of linguistic understanding, advocating for probing internal representations.

Our t-SNE visualizations corroborate this: clear linguistic clusters emerge in intermediate layers but dissolve in the final layer, reinforcing the view that the last layer is not optimal for assessing competence. These findings suggest that Direct, while more grounded than prompting, is still a limited proxy for internal knowledge.

In contrast, neurolinguistic probing inspects internal activation patterns across layers and tasks, uncovering the underlying representational structure of form and meaning, and further validates the discrepancy between performance and competence.

On the other hand, while Meta results underperform, this does not necessarily indicate that the LLMs lack the underlying linguistic competence. Instead, it may reflect limitations in information access, as suboptimal prompts can prevent models from exhibiting their full capabilities. Specifically, prompting failures do not always equate to a lack of encoded knowledge. This aligns with prior work (Firestone, 2020; Lampinen, 2024) emphasizing the need to distinguish performance conditions from underlying ability.

Thus, we argue that probing, particularly when applied layer-wise, provides a more accurate and comprehensive assessment of linguistic competence than Direct probability alone.

Form and meaning: observations from Saussure's semiotics Our results reveal that LLMs consistently learn linguistic form before they grasp meaning. This may suggest a developmental trajectory where statistical patterns in syntax and grammar are more readily captured by the model than conceptual understanding. Second, the models' formal competence is generally superior to their semantic competence. This is evident in their ability to decode grammaticality structures accurately but with less reliable conceptual accuracy.

We further observe a linear correlation between form and meaning competence, particularly when linguistic forms vary across languages while meaning remains constant. This suggests that LLMs' understanding of meaning might rely heavily on form, with conceptual representation anchored to formal structures rather than independent meaning comprehension.

These results offer a semiotic and neurolinguistic explanation for LLMs' long-standing issue of generating “confidently incorrect” responses, i.e., hallucinations (Ji et al., 2023).

6.1.7 Sum-Up

This study adopts both psycho- and neuro-linguistic approaches to evaluating LLMs, revealing a distinction between linguistic performance and competence. Our results highlight the limitations of LLMs' semantic understanding and the need for future research to move beyond statistical correlations toward more grounded language representations. By introducing a cognitive neuroscience perspective, along with semiotics, we hope will inspire further research to deepen our understanding of the language capabilities of LLMs.

6.2 Mechanistic Understanding and Mitigation of Language Confusion in English-Centric Large Language Models

This section corresponds to the following work:

> **Ercong Nie**, Helmut Schmid, and Hinrich Schuetze. 2025. Mechanistic Understanding and Mitigation of Language Confusion in English-Centric Large Language Models. In Findings of the Association for Computational Linguistics: EMNLP 2025, pages 690–706, Suzhou, China. Association for Computational Linguistics.

Declaration of Co-Authorship. I conceived the idea of using mechanistic interpretability methods to analyze and mitigate language confusion in English-centric Large Language Models (LLMs). I completed the code work and ran all the experiments. Besides, I drafted the manuscript. Helmut Schmid and Hinrich Schütze provided me with valuable feedback and advice in the process of the project.

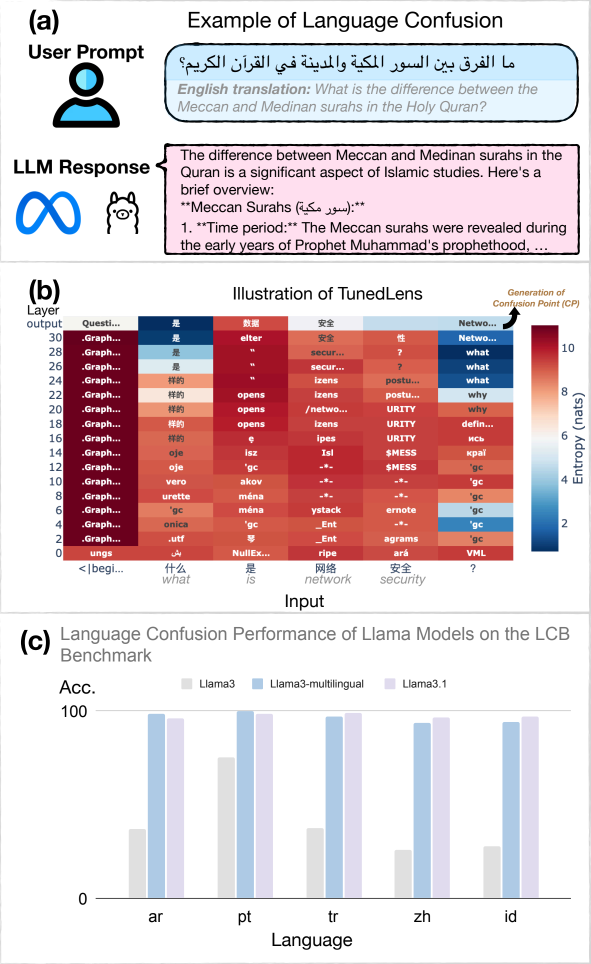
Summary of This Section

Language confusion — where large language models (LLMs) generate unintended languages against the user's need — remains a critical challenge, especially for English-centric models. We present the first mechanistic interpretability (MI) study of language confusion, combining behavioral benchmarking with neuron-level analysis. Using the Language Confusion Benchmark (LCB), we show that confusion points (CPs)—specific positions where language switches occur—are central to this phenomenon. Through layer-wise analysis with TunedLens and targeted neuron attribution, we reveal that transition failures in the final layers drive confusion. We further demonstrate that editing a small set of critical neurons, identified via comparative analysis with multilingual-tuned models, substantially mitigates confusion without harming general competence or fluency. Our approach matches multilingual alignment in confusion reduction for most languages and yields cleaner, higher-quality outputs. These findings provide new insights into the internal dynamics of LLMs and highlight neuron-level interventions as a promising direction for robust, interpretable multilingual language modeling.

6.2.1 Background and Introduction

Current Large Language Models (LLMs), such as GPT-4 (Achiam et al., 2023), PaLM 2 (Anil et al., 2023), and Llama 3 (Grattafiori et al., 2024), have demonstrated exceptional linguistic competence across a wide range of complex tasks that require abstract knowledge and reasoning (Dong et al., 2024; Wei et al., 2022b). Early LLMs were predominantly trained on massive amounts of English text data, with some limited exposure to other languages, resulting in initially constrained multilingual capabilities (Touvron et al., 2023a). Recent advances, such as multilingual continued pretraining and instruction tuning, have substantially extended these models' ability to support multiple languages (Zhu et al., 2023; Shaham et al., 2024; Kew et al., 2024; Wang et al., 2025b). As a result, contemporary English-centric LLMs have become foundational tools for multilingual communication, multilingual content generation, and cross-lingual applications (Bang et al., 2023b; Ahuja et al., 2023; Asai et al., 2024). However, despite their impressive capabilities, a persistent and underexplored limitation remains: LLMs can fail to generate text in the user's intended language, even when explicitly instructed—a phenomenon termed language confusion (Marchisio et al., 2024). Language confusion manifests as full-response, line-level, or word-level switches into unintended languages, severely undermining user experience and model reliability, especially for non-English speakers (Figure 6.12a).

Recent work by Marchisio et al. (2024) provides the first systematic characterization of language confusion, introducing the Language Confusion Benchmark (LCB) and associated metrics to quantify this phenomenon across a diverse set of languages and models. Their evaluation revealed that even state-of-the-art LLMs are susceptible to language confusion, with English-centric LLMs such as Llama2, Llama3, and Mistral exhibiting particularly high rates of unintended language switching, especially in the absence of targeted multilingual alignment (Figure 6.12c). While Marchisio et al. (2024) propose several mitigation strategies, including decoding adjustments, prompting techniques, and multilingual fine-tuning, these approaches remain largely surface-level, offering limited insight into the internal mechanisms that give rise to lan-



ما الفرق بين السور المكية والمدنية في القرآن الكريم؟

English translation: What is the difference between the Meccan and Medinan surahs in the Holy Quran?

The difference between Meccan and Medinan surahs in the Quran is a significant aspect of Islamic studies. Here's a brief overview:
Meccan Surahs (سور مكية):
1. **Time period:** The Meccan surahs were revealed during the early years of Prophet Muhammad's prophethood, …

Layer	<|begi...	什么 what	是 is	网络 network	安全 security	?
output	Questi...	是	数据	安全		Netwo...
30	.Graph...	是	elter	安全	性	Netwo...
28	.Graph...	是	"	secur...	?	what
26	.Graph...	是	"	secur...	?	what
24	.Graph...	样的	"	izens	postu...	what
22	.Graph...	样的	opens	izens	postu...	why
20	.Graph...	样的	opens	/netwo...	URITY	why
18	.Graph...	样的	opens	izens	URITY	defin...
16	.Graph...	样的	ę	ipes	URITY	ись
14	.Graph...	oje	isz	Isl	$MESS	краї
12	.Graph...	oje	'gc	-*-	$MESS	'gc
10	.Graph...	vero	akov	-*-	-*-	'gc
8	.Graph...	urette	ména	-*-	-*-	'gc
6	.Graph...	'gc	ména	ystack	ernote	'gc
4	.Graph...	onica	'gc	_Ent	-*-	'gc
2	.Graph...	.utf	琴	_Ent	agrams	'gc
0	ungs	بش	NullEx...	ripe	ará	VML

(c)
Language Confusion Performance of Llama Models on the LCB Benchmark
Acc.
Llama3
Llama3-multilingual
Llama3.1
100
0
ar
pt
tr
zh
id
Language

Figure 6.12: Language Confusion in LLMs. (a) An example of the language confusion phenomenon. (b) Visualization of internal model dynamics using TunedLens, highlighting how the confusion point emerges during generation. (c) Benchmarking results of three Llama models on the LCB benchmark across 5 languages.

guage confusion.

A key observation from prior work is the identification of confusion points—specific positions in the generation process where the model abruptly switches to an unintended language. However, the model's internal dynamics leading to these confusion points and their causal role in language confusion remain largely unexplored. This gap is particularly salient given the parallels to human bilingual code-switching, where switch points between languages are cognitively significant as extensively studied in psycholinguistics (Solorio and Liu, 2008; Bullock and Toribio, 2009).

In this work, we move beyond behavioral evaluation to open the black box of LLMs, leveraging mechanistic interpretability (MI) methods (Conmy et al., 2023; Rai et al., 2024; Saphra and Wiegreffe, 2024; Sharkey et al., 2025) to investigate the internal representations and neuron-level processes underlying language confusion. We first empirically demonstrate that confusion points are critical drivers of language confusion: targeted interventions at these points can substantially reduce confusion across languages. Building on this, we employ MI tools such as *TunedLens* (Belrose et al., 2023) to trace the evolution of language representations through the model's layers, revealing that confusion typically arises from transition failures in the final layers, where latent conceptual representations are mapped to surface forms in the target language (Figure 6.12b). To further elucidate the mechanism, we conduct a neuron-level analysis, identifying specific neurons in the last layers whose activity is predictive of successful or failed language transitions at confusion points. Inspired by recent advances in neuron attribution and editing, we show that targeted manipulation of only 100 neurons can mitigate language confusion, offering a novel, model-internal approach to improving multilingual reliability. Our findings provide the first mechanistic account of language confusion in LLMs, bridging the gap between behavioral benchmarks and internal model dynamics. By highlighting the central role of confusion points and their neural substrates, we lay the groundwork for more robust, interpretable, and cognitively informed multilingual language models.

Our work makes the following contributions: (1) We provide the first mechanistic interpretability study of language confusion in English-centric LLMs, revealing the central role of confusion points in unintended language switching; (2) We employ layer-wise and neuron-level analyses to trace the internal dynamics leading to language confusion and identify critical late-layer neurons responsible for transition failures; (3) We propose and validate a principled neuron selection and editing strategy that effectively mitigates language confusion and preserves the model's general competence and output quality.

6.2.2 Language Confusion and Mechanistic Interpretability

Code-switching as a Linguistic Phenomenon Code-switching, the practice of alternating between languages within a single conversation or utterance, is a well-studied natural phenomenon in bilingualism and psycholinguistics (Gardner-Chloros, 2009). Code-switching is typically intentional, often reflecting speakers' identities, social relationships, and contextual adaptation (Treffers-Daller, 2009; Yim and Clément, 2021). In NLP, code-switching has been explored through evaluating model performance on code-switched data for tasks such as sentiment analysis, machine translation, summarization, and language identification (Khanuja et al., 2020; Doğruöz et al., 2021; Winata et al., 2023). Code-switching is a natural, contextually appropriate strategy in human communication, whereas language confusion, on which our work focuses, is an unintended and erroneous switch to an incorrect language in LLMs (Marchisio et al., 2024). Though related to code-switching, language confusion is an unnatural phenomenon that arises from model failures rather than communicative intent.

Language Confusion and Confusion Points in LLMs Language confusion has been observed in various multilingual NLP settings, such as "source language hallucinations" in zero-shot cross-lingual transfer (Li and Murray, 2023; Pfeiffer et al., 2023; Chirkova and Nikoulina, 2024) and "off-target translation" in machine translation (Sennrich et al., 2024). In LLMs, this manifests as abrupt, unexpected switches to the wrong language during generation, even under explicit instructions. This issue is particularly prevalent in English-centric models lacking robust multilingual alignment (Zhong et al., 2024). A key concept in recent work is the *confusion point*—the specific position in generation where the model transitions to an unintended language. Inspired by the importance of code-switching points in human bilingualism, confusion points are central to understanding and diagnosing language confusion in LLMs (Guzzardo Tamargo et al., 2016). Unlike natural code-switching, these points reflect internal model failures. Recent benchmarks (Marchisio et al., 2024) systematically characterize confusion points at response, line, and word levels, revealing their widespread impact and motivating deeper mechanistic investigation, as pursued in this work.

Mechanistic Interpretability Methods Mechanistic interpretability (MI) seeks to reverse-engineer neural networks by decomposing their computations into human-understandable components (Stolfo et al., 2023; Wang et al., 2024b; Men et al., 2024). A central technique in MI is the projection of intermediate representations into the vocabulary space, as implemented by tools such as LogitLens (Nostalgebraist, 2020) and TunedLens (Belrose et al., 2023), which enable researchers to track how information and predictions evolve across layers (Dar et al., 2023; Pal et al., 2023). In addition to layer-wise analysis, recent work has focused on identifying, attributing, and intervening on important neurons—those whose activations are strongly correlated with specific linguistic functions or behaviors (Bau et al., 2020; Geva et al., 2022; Yu and Ananiadou, 2024b). Methods for neuron selection and editing, as well as circuit-level analysis (Elhage et al., 2021; Wang et al., 2023b), have proven effective for uncovering the internal structure underlying phenomena such as factual recall (Meng et al., 2022; Geva et al., 2023), reasoning processing (Yu and Ananiadou, 2024a), and now, as in our work, language confusion. By leveraging these MI techniques, we aim to provide a granular, causal understanding of how and why language confusion arises in multilingual LLMs, and to identify actionable intervention points for mitigation.

Multilingual Interpretability Recent research has begun to probe the internal representations of English-centric and multilingual LLMs to understand how they process and transfer information across languages (He et al., 2024b; Zhao et al., 2024). Wendler et al. (2024) show that models like Llama2 often rely on English as an internal pivot language and can disentangle language and conceptual representations in controlled tasks. Fierro et al. (2025) examine how mechanisms identified in monolingual contexts generalize to multilingual settings. Wang et al. (2025a) investigate the internal causes of crosslingual factual inconsistencies, revealing how MLMs transition from language-independent to language-specific processing. However, prior work has not systematically connected these internal mechanisms to language generation errors such as language confusion.

Dataset	Data Source	Language	Prompt Example
Aya (Singh et al., 2024)	Human-generated	ar, en, pt, tr, zh	请简单介绍诗人李白的背景。 *Briefly introduce the poet Li Bai.*
Dolly (Singh et al., 2024)	MT post-edited	ar, es, fr, hi, ru	Qu'est-ce qui est plus important, l'inné ou l'acquis? *What is more important, nature or nurture?*
Native (Marchisio et al., 2024)	Human-generated	es, fr, ja, ko	콘크리트는 뭘로 만든거야? *What is concrete made of?*
Okapi (Lai et al., 2023b)	Synthetic + MT	ar, en, pt, zh,it, fr, de, id, es, vi	Schreib einen Aufsatz von 500 Wörtern zum Thema KI. *Write a 500-word essay on AI.*

Table 6.4: Overview and Prompt Example of the LCB Benchmark (monolingual part). The number of examples per language is 100 in each dataset.

6.2.3 Revisiting Language Confusion: Benchmark Insights

Recap of Language Confusion Benchmark The Language Confusion Benchmark (LCB) (Marchisio et al., 2024) provides a systematic framework for evaluating the ability of LLMs to generate text in the user's intended language. The benchmark covers 15 typologically diverse languages and uses a diverse set of prompts sourced from human-written, post-edited, and synthetic datasets to evaluate models, ensuring coverage of a wide range of domains and linguistic structures (Table 6.4). In this work, we focus on the *monolingual setting* of LCB, where the prompt and expected response are in the same language. This setting is particularly relevant for mechanistic interpretability research, as it isolates language confusion phenomena from the additional complexities of explicit cross-lingual transfer.

To quantify language confusion, we adopt two key metrics from LCB: **line-level pass rate (LPR)** and **line-level language accuracy (Acc)**. LPR measures the percentage of model responses in which every line is in the correct language. Acc reflects the proportion of individual lines across all responses that are correctly generated in the target language. Both metrics rely on automatic language identification using the fastText classifier (Joulin et al., 2016, 2017), which efficiently detects the language of each line in the generated output.

We conducted preliminary benchmarking experiments on LCB with three instruction-tuned LLMs: ***Llama3-8B*** *(English-centric, no multilingual instruction tuning)*, ***Llama3-8B-multilingual*** *(multilingual instruction-tuned)* (Devine, 2024), and ***Llama3.1-8B*** *(multilingual-optimized)*. As shown in Figure 6.12c, *Llama3-8B* exhibits substantial language confusion, with frequent line-level switches to unintended languages (mostly English). In contrast, both *Llama3-8B-multilingual* and *Llama3.1-8B* achieve near-perfect LPR and line-level accuracy, demonstrating the effectiveness of multilingual instruction tuning and targeted optimization for multilingual dialogue.

Given these findings, our work centers on understanding and mitigating the language confusion observed in English-centric *Llama3-8B*. By leveraging mechanistic interpretability methods, we aim to uncover the internal causes of confusion and develop interventions that can bring its performance closer to that of explicitly multilingual-tuned models. In the following subsection, we delve deeper into the significance of confusion points as critical junctures in the generation process.

Model	**Metric**	ar	en	pt	tr	zh	es	fr	hi	ru	ja	ko	de	id	it	vi	**avg**
Llama3	*LPR*	33.0	99.5	71.0	33.0	19.3	73.0	59.3	8.0	28.0	14.0	23.0	19.0	22.0	34.0	11.0	**36.5**
(original)	*Acc*	33.7	99.8	74.5	37.5	23.4	77.1	64.1	15.1	28.2	17.1	23.6	23.0	27.3	39.8	14.8	**39.9**
Llama3	*LPR*	71.0	99.0	93.0	50.0	57.3	94.3	84.0	37.0	78.6	50.0	45.0	60.0	67.0	86.0	62.0	**68.9**
(replace)	Acc	74.8	99.6	95.4	55.5	64.1	95.3	86.5	47.6	83.1	55.3	48.6	62.3	77.7	87.5	66.1	**73.3**
Llama3	*LPR*	98.3	98.5	99.0	95.8	88.8	98.3	95.9	97.0	100.0	93.5	100.0	100.0	88.8	100.0	97.9	**96.8**
(multilingual)	*Acc*	98.7	99.5	99.8	96.9	93.8	99.3	96.9	97.5	100.0	95.8	100.0	100.0	94.2	100.0	97.9	**98.0**

Table 6.5: Impact of Confusion Point Replacement on Language Confusion Metrics. Line-level pass rate (LPR) and line-level accuracy for original Llama3-8B, multilingual Llama3-8B, and Llama3-8B with confusion point replacement, reported by language.

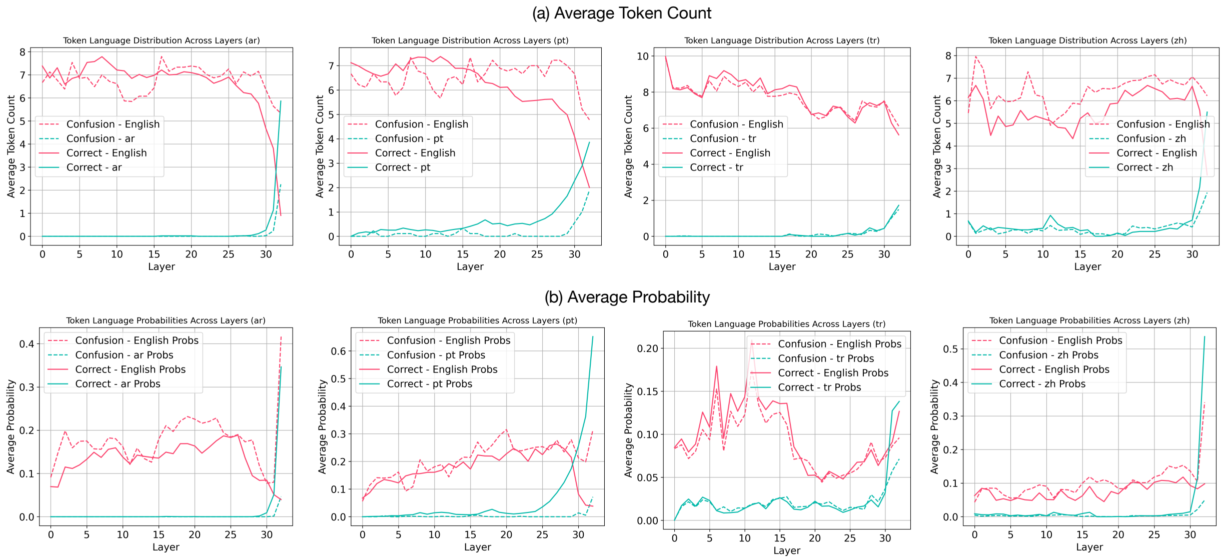
Significance of Confusion Points A confusion point (CP) is the position in a model's output where the first token of an unintended language abruptly appears, marking the onset of language confusion (Marchisio et al., 2024). This concept is inspired by psycho- and neurolinguistic research on code-switching, where the precise location of a language switch—known as a switch point—is central to understanding bilingual language production and processing (Blanco-Elorrieta and Pylkkänen, 2017; Suurmeijer et al., 2020). To empirically assess the role of CPs in LLM language confusion, we conduct a replacement experiment on *Llama3-8B*. For each instance of language confusion, we identify the CP using the fastText language detector. We then replace the token at the CP with the corresponding token generated by *Llama3-8B-multilingual*, which achieves near-perfect language accuracy, under the same prompt. This approach is motivated by the psycholinguistic observation that, in human code-switching, the choice at the switch point strongly influences the subsequent language trajectory (Moreno et al., 2002; Lai and O'Brien, 2020).

6.2.4 Mechanistic Analysis of Language Confusion Points

6.2.4.1 Analyzing Layer-wise Language Transition

A central question in understanding language confusion is where and how the model's internal representations fail to transition from a shared conceptual space to the intended target language. Motivated by recent findings that English-centric LLMs process information in a latent, often English-biased, conceptual space before converting it to the target language in the final layers (Wendler et al., 2024; Wang et al., 2025a), we conduct a detailed layer-wise analysis of this transition using TunedLens (Belrose et al., 2023).

We employ TunedLens, the more reliable variant of LogitLens (Nostalgebraist, 2020), to unembed the hidden states of *Llama3-8B* at each layer into the vocabulary space. With this, we inspect every layer of the model and extract the top 10 predicted tokens with the largest logits at the position immediately preceding the confusion point (CP) (for confusion cases) or the output token (for correct cases). For each layer, we compute the average number and summed probabilities of English and target language tokens among the top-10 predictions, using fastText for language identification. Our analysis focuses on four typologically diverse languages (Arabic, Portuguese, Turkish, Chinese) from the LCB benchmark. We separate samples into two groups: (1) *Correct*—where the model generates the intended language throughout, and (2) *Confusion*—


(a) Average Token Count

(b) Average Probability

Figure 6.13: Average token counts and probabilities for English and target language tokens among the top-10 predictions at each layer, shown for both correct and confusion samples across four languages from *Aya*.

where the model switches to an unintended language at a CP. For confusion samples, we analyze the model's state up to the token before the CP.

Figure 6.13 presents the evolution of language token counts and probabilities across layers for both groups. In early and middle layers, English tokens dominate the top-10 predictions for all languages, reflecting the English-centric latent conceptual space of *Llama3-8B*. This is consistent with prior work showing that LLMs encode information in a shared, language-agnostic space in intermediate layers. In the final layers, a sharp transition emerges. For correct samples, the number and probability of target language tokens rise steeply, overtaking English tokens in the last few layers—indicating a successful transition to the target language surface form. In contrast, for confusion samples, this transition fails: English tokens remain dominant or even increase, while target language tokens lag behind. This failure to shift from the latent conceptual space to the target language at the critical moment leads to CPs and erroneous output.

Our layer-wise analysis with TunedLens reveals that the transition to the target language occurs in the final layers, and that failures in this process are tightly linked to language confusion. These findings provide direct evidence that language confusion in *Llama3-8B* is primarily caused by transition failures in the last few layers, motivating our subsequent neuron-level investigation to pinpoint and intervene on the specific components responsible for these failures.

6.2.4.2 Localizing Critical Neurons at Confusion Points

A key step toward understanding and mitigating language confusion is to identify which neurons are most responsible for the emergence of confusion points. Building on recent advances in neuron-level attribution (Geva et al., 2022; Yu and Ananiadou, 2024b), we adopt a static, efficient method to locate and analyze the most influential feed-forward network (FFN) neurons in *Llama3-8B*.

Methodology In the inference pass in decoder-only LLMs, for a given input sequence, each layer output h_i^l (layer l, token position i) is a sum of the previous layer's output h_i^{l-1}, the attention output A_i^l, and the FFN output F_i^l:

$$h_i^l = h_i^{l-1} + A_i^l + F_i^l \tag{6.2}$$

The FFN output F_i^l is calculated with a non-linear activation function σ and two feedforward layers $W_{fc1}^l \in \mathbb{R}^{N \times d}$ and $W_{fc2}^l \in \mathbb{R}^{d \times N}$:

$$F_i^l = W_{fc2}^l \sigma(W_{fc1}^l (h_i^{l-1} + A_i^l)) \tag{6.3}$$

Following Geva et al. (2021), the FFN layer output F_i^l can be represented as a weighted sum over neuron subvalues:

$$F_i^l = \sum_{k=1}^{N} m_{i,k}^l \cdot fc2_k^l \tag{6.4}$$

$$m_{i,k}^l = \sigma(fc1_k^l \cdot (h_i^{l-1} + A_i^l)) \tag{6.5}$$

where $fc2_k^l$ is the k-th column of W_{fc2}^l, and $m_{i,k}^l$ is derived from the inner product between the residual output $(h_i^{l-1} + A_i^l)$ and $fc1_k^l$, the k-th row of W_{fc1}^l.

Geva et al. (2022) and Dar et al. (2023) project FFN neuron subvalues with unembedding matrices to compute the token probability distribution. To quantify the importance of each neuron for generating a specific token (e.g., at a confusion point), we adopt the log probability increase method of Yu and Ananiadou (2024b). For a neuron in the l-th FFN layer v^l, its importance score is defined as the increase in log probability of the target token when v^l is added to the residual stream $A^l + h^{l-1}$, compared to the baseline without v^l:

$$Imp(v^l) = \log(p(w|v^l + A^l + h^{l-1}) - \log(p(w|A^l + h^{l-1}) \tag{6.6}$$

This approach efficiently identifies neurons whose activations most strongly influence the model's prediction at a given position.

Experimental Observations We apply this method to *Llama3-8B* on confusion samples from the LCB benchmark, focusing on the token position immediately preceding each confusion point. For each sample and language, we compute the importance scores for all 14,336 FFN neurons in each layer of *Llama3-8B*, rank them, and select the top 300 most important neurons per sample. We then analyze the distribution of these critical neurons across layers, both for individual samples and aggregated over all samples in a language. Our analysis reveals a striking concentration of important neurons in the final layers, as visualized in Figure 6.14. This pattern holds both at the single-sample level and when aggregating across samples, indicating that the emergence of confusion points is primarily driven by late-layer FFN activity. We further rank neurons by

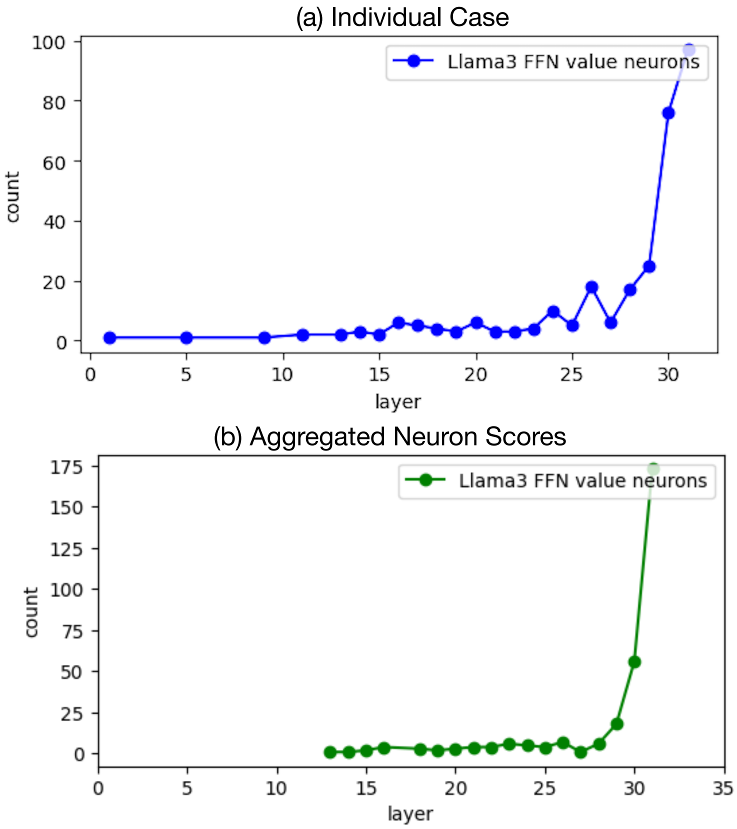


Figure 6.14: Distribution of Important Neurons Associated with Confusion Points in *Llama3-8B*. (a) Distribution of the top 300 most important FFN neurons across layers for an individual Chinese prompt "请解释拆东墙补西墙的意思。*(Please explain '拆东墙补西墙.')*" from Aya. (b) Aggregated distribution of important neuron scores across all Chinese test samples in Aya.

their frequency of appearance in the top 300 sets across samples, finding that a subset of neurons consistently recurs as highly influential for confusion points.

To understand the effect of multilingual alignment, we repeat the analysis on *Llama3-8B-multilingual* using the same set of prompts. After multilingual instruction tuning, language confusion is nearly eliminated. Comparing neuron importance scores between the two models (Figure 6.15), we observe that most neurons critical for confusion in the *Llama3-8B* become much less important in its multilingual counterpart, suggesting that multilingual alignment suppresses the activity of confusion-inducing neurons. However, a small number of neurons remain important or even increase in importance, likely reflecting their role in encoding general semantic information rather than language-specific transitions.

These findings reinforce the conclusion from our layer-wise analysis: language confusion is tightly linked to the activity of specific FFN neurons in the *final* layers. The suppression of these neurons through multilingual alignment provides a mechanistic explanation for the effectiveness of such tuning. Moreover, the identification of a small set of persistent, semantically important neurons suggests that *targeted* neuron-level interventions could mitigate confusion without

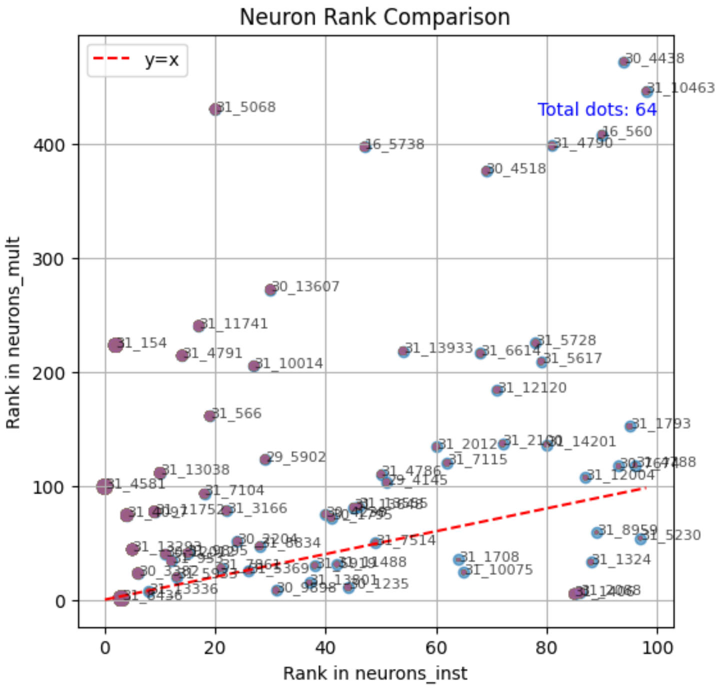


Figure 6.15: Neuron rank comparison between original Llama3 and multilingual Llama3. Results of Chinese test samples in Aya.

harming overall model performance. These insights directly inform our subsequent strategies for neuron-based mitigation of language confusion.

6.2.5 Mitigating Language Confusion via Neuron Editing

A central challenge in mitigating language confusion via neuron editing is to identify a set of neurons whose intervention effectively reduces confusion without degrading the model's general competence or fluency. Insights from our previous mechanistic analysis indicate that language confusion is primarily driven by a subset of late-layer FFN neurons. However, indiscriminate deactivation of important neurons risks harming the model's overall performance. Thus, a principled neuron selection strategy is essential.

6.2.5.1 Neuron Selection and Intervention

We compare three neuron selection strategies: (1) *Frequency-Based Selection:* Selects the neurons most frequently identified as important across all confusion samples for a given language. (2) *Aggregate Importance Selection:* Ranks neurons by the sum of their importance scores across all confusion samples, selecting those with the highest cumulative influence. While this method captures the overall impact, it may still include neurons essential for general language competence. (3) *Comparative Importance Selection:* Inspired by Yu and Ananiadou (2024a), this strategy identifies neurons whose importance scores for confusion points decrease most substantially after multilingual alignment. Specifically, for each neuron, we compute the difference in importance score between original *Llama3-8B* and *Llama3-8B-multilingual* on the same input. Neurons with the largest drop are prioritized for intervention, as they are likely to be specifically implicated in language confusion rather than general semantic processing.

	ar	pt	tr	zh	es	fr	hi	ru	ja	ko	de	id	it	vi	**Avg.**
original	33.44	74.26	37.55	24.04	77.15	63.16	16.47	28.20	17.44	23.50	23.00	27.33	39.83	14.79	**35.73**
freq	31.75	75.10	36.51	22.09	76.29	66.98	18.66	27.70	19.29	23.08	22.25	27.83	39.45	13.58	**35.75**
score	76.97	93.41	67.61	80.63	91.22	74.77	60.00	50.32	53.50	33.25	40.27	53.58	96.00	67.56	**67.08**
comparative	85.45	97.12	57.27	89.39	92.20	83.17	82.74	89.43	49.95	40.33	80.82	78.94	95.25	66.50	**77.75**

Table 6.6: Confusion mitigation performance of different selection strategies. Line-level accuracy is reported.

For each strategy, we select the top 100 neurons and intervene by setting their activations to zero during generation. We evaluate the impact of each method on the LCB benchmark. Our results (Table 6.6) demonstrate that Comparative Importance Selection achieves the most effective reduction in language confusion, substantially outperforming both frequency-based and aggregate importance methods. Frequency-based selection yields minimal benefit, while aggregate importance provides moderate improvement but still lags behind our proposed approach. Notably, the comparative strategy selectively targets neurons implicated in confusion, minimizing collateral impact on general model competence.

6.2.5.2 Generalization and Robustness of Neuron Editing

To further validate the effectiveness and safety of our Comparative Importance Selection strategy, we conduct a comprehensive evaluation across multiple metrics and experimental setups. Our goal is to ensure that neuron editing not only mitigates language confusion but also preserves the model's general competence, fluency, and robustness across domains (Table 6.7).

	token_num	token_prob	fluency	acc_ood	xnli	senti
Original	1.96	24.5	25.8	39.9	46.4	98.4
Edited	3.43	36.8	21.8	74.25	44.9	98.2
Diff	1.47	12.3	-4.0	34.4	-1.5	-0.2

Table 6.7: Results of generalization and robustness of neuron editing. Average performance across languages is reported. Detailed results in Appendix H.

Language Confusion Mitigation We first assess the impact of neuron editing on language confusion using the LCB benchmark. In addition to standard metrics (line-level pass rate and line-level accuracy), we analyze the internal output distributions by reporting (1) the number of target language tokens among the top-10 candidates in the final output token logit, and (2) the total probability mass assigned to target language tokens in the top-10. These metrics provide a deeper view of how neuron editing shifts the model's internal preference toward the intended language, beyond surface-level accuracy.

Robustness on General Tasks To evaluate whether neuron editing affects the model's general capabilities, we test the edited model on widely used multilingual benchmarks, including XNLI and multilingual sentiment analysis. We also assess output fluency by measuring the perplexity of generated responses using the multilingual model `facebook/xglm-564M` (Lin et al., 2022). Across all these metrics, the edited model maintains performance comparable to the original Llama3-8B, indicating that our intervention does not degrade general language understanding or generation quality.

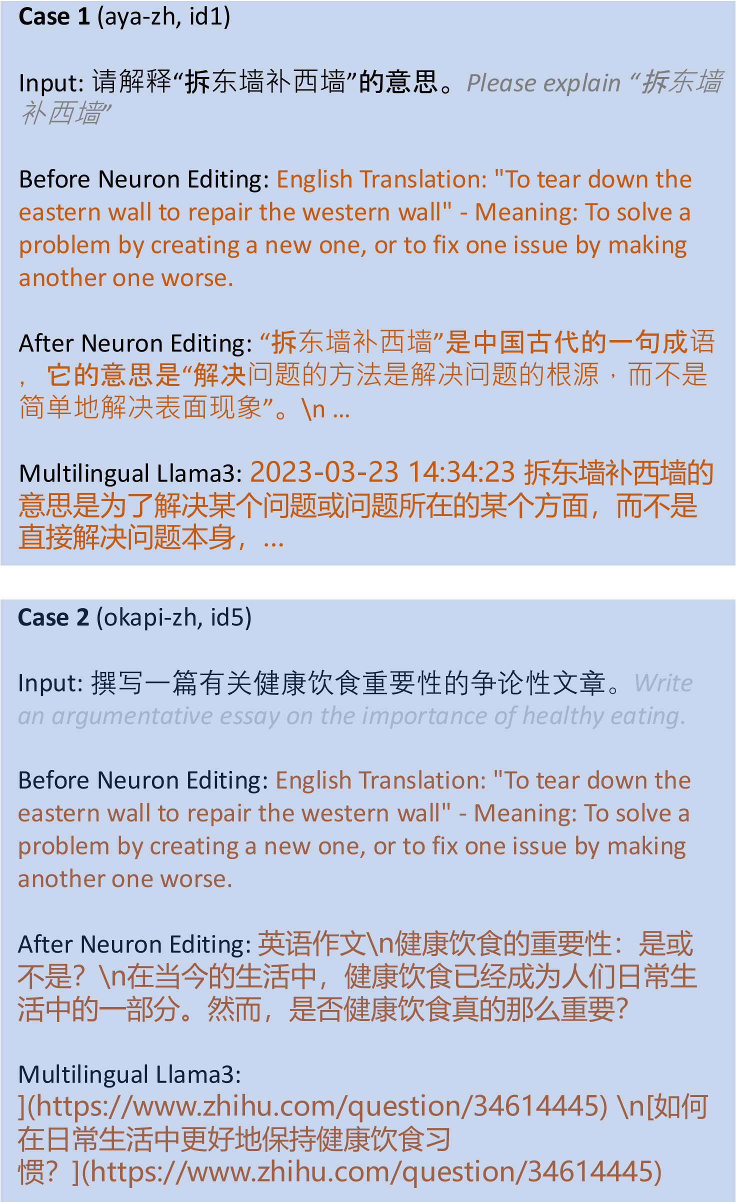
Out-of-Domain Generalization We further examine the generalization of neuron editing by applying neurons selected from one data source (e.g., Aya) to out-of-domain test sets (e.g., Okapi) for the same language. The edited model continues to demonstrate strong mitigation of language confusion, suggesting that the identified neurons capture robust, domain-independent mechanisms underlying confusion points.

6.2.5.3 Comparison with Multilingual Alignment

To contextualize the effectiveness of neuron editing, we compare the performance of the edited *Llama3-8B* model with that of the multilingual-tuned *Llama3-8B*. Quantitative results show that neuron editing achieves language confusion mitigation on par with the multilingual-aligned model for most languages. Both approaches yield high line-level pass rates and language accuracy, demonstrating that targeted neuron intervention can match the benefits of extensive multilingual instruction tuning for confusion reduction.

Beyond aggregate metrics, qualitative analysis reveals further advantages of the neuron editing approach. Through case studies, we observe that the edited Llama3-8B not only generates fluent and accurate responses in the intended target language, but also avoids certain artifacts introduced by multilingual alignment. For example, in several instances (Figure 6.16), the multilingual Llama3-8B produces outputs containing HTML tags or formatting patterns reminiscent of its instruction tuning data, reflecting the influence of imperfect or noisy multilingual datasets. In contrast, the neuron-edited model consistently produces clean, well-structured, and contextually appropriate responses, free from such extraneous formatting.

These findings highlight a key strength of mechanistic neuron editing: it directly addresses the internal causes of language confusion without introducing side effects from large-scale data-driven alignment. By preserving the original model's semantic competence and output quality, neuron editing offers a more targeted and interpretable solution. This suggests that, beyond traditional multilingual instruction tuning, mechanistic interpretability-driven interventions can provide a promising path toward high-quality, robust multilingual language models.



Case 1 (aya-zh, id1)

Input: 请解释“拆东墙补西墙”的意思。*Please explain “拆东墙补西墙”*

Before Neuron Editing: English Translation: "To tear down the eastern wall to repair the western wall" - Meaning: To solve a problem by creating a new one, or to fix one issue by making another one worse.

After Neuron Editing: “拆东墙补西墙”是中国古代的一句成语，它的意思是“解决问题的方法是解决问题的根源，而不是简单地解决表面现象”。\n ...

Multilingual Llama3: 2023-03-23 14:34:23 拆东墙补西墙的意思是为了解决某个问题或问题所在的某个方面，而不是直接解决问题本身，...

Case 2 (okapi-zh, id5)

Input: 撰写一篇有关健康饮食重要性的争论性文章。*Write an argumentative essay on the importance of healthy eating.*

Before Neuron Editing: English Translation: "To tear down the eastern wall to repair the western wall" - Meaning: To solve a problem by creating a new one, or to fix one issue by making another one worse.

After Neuron Editing: 英语作文\n健康饮食的重要性：是或不是？\n在当今的生活中，健康饮食已经成为人们日常生活中的一部分。然而，是否健康饮食真的那么重要？

Multilingual Llama3:](https://www.zhihu.com/question/34614445) \n[如何在日常生活中更好地保持健康饮食习惯？](https://www.zhihu.com/question/34614445)

Figure 6.16: Case study of neuron editing.

6.2.6 Sum-Up

This work provides the first mechanistic interpretability account of language confusion in English-centric LLMs. By tracing confusion points to failures in late-layer transitions and localizing the critical neurons responsible, we demonstrate that targeted neuron editing can robustly mitigate language confusion without sacrificing general competence or fluency. Our approach achieves results on par with multilingual-tuned models for most languages, while preserving cleaner output quality. These findings highlight the promise of neuron-level interventions for more reliable and interpretable multilingual language modeling.

Chapter 7

Conclusion

7.1 Summary of Research

This dissertation has explored new frontiers in efficient and human-inspired natural language processing (NLP) for multilingual and low-resource settings. Motivated by the global challenge of language inequality, the work addressed the dual imperative of developing practical, scalable methods that extend NLP's reach to underrepresented languages, while simultaneously deepening our scientific understanding of language models through human-inspired analysis and interpretability.

The research presented in this dissertation can be grouped into four main threads:

- **Prompt-Based Learning for Multilingual Prediction:** The dissertation advanced prompt-based learning methods for multilingual NLP, with a focus on zero- and few-shot scenarios where labeled data is scarce. By investigating and mitigating bias in prompt-based models through calibration techniques and introducing retrieval-augmented prompting (PARC), the work achieved robust improvements in multilingual performance, particularly for low-resource and typologically diverse languages. Decomposed prompting strategies enabled more granular evaluation of linguistic structure knowledge in large language models (LLMs), while the BMIKE-53 benchmark extended prompt-based learning to the challenging domain of cross-lingual knowledge editing.

- **Prompt-Based Fine-Tuning for Cross-Lingual Transfer:** Building on the prompt-based foundation, the dissertation systematically compared prompt-based fine-tuning with traditional fine-tuning for zero-shot cross-lingual transfer. Through the PROFIT pipeline, it was shown that prompt-based fine-tuning consistently yields superior transfer, especially in low-data regimes and for typologically similar languages. The ToPro methodology extended this success to token-level tasks like POS tagging and NER, and a delexicalized constituency parser demonstrated the feasibility of cross-lingual transfer for historical languages (e.g., Middle High German), opening new avenues in computational historical linguistics.

- **Efficient NLP Methods for Low-Resource Settings:** Recognizing the practical bottlenecks of data and computational resource constraints, the dissertation introduced AMD^{2}G, a unified data augmentation framework for multi-domain dialogue generation, and GNNavi, a parameter-efficient fine-tuning method based on graph neural networks. Both approaches significantly improved performance in low-resource scenarios, demonstrating that efficiency and scalability can be achieved without sacrificing much accuracy or inclusivity.

- **Human-Inspired Understanding and Mechanistic Interpretability:** The final research thread shifted from performance to understanding, introducing probing methods inspired by psycholinguistics and neurolinguistics to distinguish between LLMs' performance and true linguistic competence. Minimal pair probing and cross-lingual analysis revealed that LLMs encode linguistic form more robustly than meaning, and that instruction tuning improves performance but not the underlying competence. Mechanistic interpretability techniques traced failure modes such as language confusion to a small set of late-layer neurons, and demonstrated that targeted neuron-level interventions could robustly mitigate these errors without harming general model competence or output quality.

7.2 Discussions and Insights

Several key insights emerge from the work presented in this dissertation:

Regarding prompt-based learning for multilingual prediction:

- **Prompt-based methods, when carefully calibrated and augmented, are highly effective for zero- and few-shot multilingual prediction.** Probability calibration and cross-lingual retrieval-augmented prompting enable models to overcome label bias and leverage information from high-resource languages, substantially boosting performance for underrepresented languages.

Regarding prompt-based fine-tuning for cross-lingual transfer:

- **Prompt-based fine-tuning offers consistent advantages over vanilla fine-tuning for cross-lingual transfer,** especially in low-resource and typologically similar scenarios. Token-level decomposition (ToPro) further extends these benefits to structured prediction tasks, and the delexicalization constituency parsing research on Middle High German demonstrates strong transfer even to historical languages.

Regarding efficient NLP methods for low-resource settings:

- **Unified data augmentation and parameter-efficient adaptation are crucial for practical deployment of NLP in low-resource settings.** The AMD^{2}G and GNNavi frameworks show that it is possible to achieve competitive results with minimal data and parameter updates, making NLP more accessible and sustainable.

Regarding human-inspired understanding and mechanistic interpretability:

- **Human-inspired probing, combining psycholinguistic and neurolinguistic paradigms, reveals a gap between model performance and underlying competence.** LLMs more easily acquire linguistic form than meaning, and instruction tuning improves performance but not deep understanding. This underscores the importance of interpretability and cognitive alignment in the next generation of NLP systems.

- **Mechanistic interpretability can identify and address specific failure modes,** such as language confusion, through targeted neuron-level interventions. This opens up new possibilities for efficient, interpretable, and robust multilingual language models.

7.3 Outlook and Future Directions

While this dissertation has made significant advances in both the practical and scientific dimensions of multilingual and low-resource NLP, several promising directions for future research emerge:

- **Culturally and Socially Aware Multilingual Language Modeling:** The next generation of multilingual language models should move beyond linguistic diversity to embrace cultural and social context. This includes developing culturally sensitive prompts, datasets, and evaluation metrics, as well as modeling the diverse conceptualizations and communicative norms that arise in different linguistic and cultural communities. Future work could systematically investigate how LLMs encode, transfer, and sometimes misrepresent cultural knowledge, and develop methods for cross-cultural calibration and alignment.

- **Cross-Cultural and Cross-Lingual Conceptual Understanding:** Extending multilingual interpretability research, future studies could probe how LLMs represent and process shared and divergent conceptual structures across languages and cultures. This could involve developing new benchmarks for cross-lingual conceptual alignment, investigating the mechanisms that support or hinder shared understanding across linguistic boundaries, and identifying where models fail to capture culturally specific distinctions. Such research could illuminate both the universals and particulars of human communication as reflected in language models.

- **Human-Inspired and Neuro-Cognitive Modeling for Multilingual NLP:** The integration of insights from psycholinguistics, neurolinguistics, and cognitive science holds great promise for advancing both model design and interpretability. Future research could leverage mechanistic findings from LLMs to inspire new hypotheses about human language processing and disorders, or vice versa. For example, identifying neuron-level circuits responsible for language confusion in LLMs may inspire analogous investigations in bilingual aphasia or code-switching in the human brain. Conversely, cognitive models of multilingual acquisition and processing can inform the architecture and training of more human-like multilingual models.

- **Towards Human-Centric, Inclusive, and Ethical NLP:** As NLP technologies continue to shape global communication, it is imperative that future research prioritizes fairness, inclusivity, and the ethical use of language models. This includes the responsible collection and annotation of culturally diverse data, privacy-preserving deployment in sensitive contexts, and continuous evaluation of social biases and unintended consequences in real-world use. Embedding human-centric values in both the design and evaluation of multilingual NLP systems will be critical for ensuring that technological progress benefits all.

In summary, this dissertation has contributed new methods, analyses, and perspectives for efficient and human-inspired multilingual NLP, bridging the gap between practical performance and scientific understanding. By combining algorithmic innovation with cognitive and cultural insight, it lays the groundwork for future research that is not only technologically advanced, but also more inclusive, interpretable, and attuned to the complexity of human language and society.

Appendix

A Detailed PARC Results

We show the detailed experimental results for all tasks in Table 1 (Amazon reviews), Table 2 (AG News), and Table 3 (XNLI), respectively.

pattern 0	[X] [MASK]
pattern 1	It was [MASK]. [X]
pattern 2	[X] All in all, it was [MASK].
pattern 3	Just [MASK]! [X]
pattern 4	[X] In summary, the product is [MASK].

				en					**af**					**ur**		
		p0	p1	p2	p3	p4	p0	p1	p2	p3	p4	p0	p1	p2	p3	p4
MAJ		50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
Direct		50.5	54.3	58.9	53.7	52.6	53.3	50.7	50.4	49.8	51.5	49.9	51.7	54.6	49.9	50.3
Unlabeled	k=1	**50.9**	**55.4**	**59.1**	**51.9**	**52.6**	**51.0**	**54.9**	**57.9**	**52.9**	**52.8**	**51.6**	**56.7**	**60.0**	**52.2**	**52.2**
	k=3	50.7	53.7	57.7	50.8	50.4	50.4	52.5	56.2	50.7	51.0	51.3	52.9	57.1	50.8	50.9
	k=5	50.8	52.2	56.0	50.3	50.9	50.8	52.2	55.0	50.2	50.6	51.2	52.5	56.4	50.3	50.7
	k=10	50.7	51.9	56.0	50.0	50.6	50.7	52.0	55.8	50.2	50.7	51.4	52.4	55.5	50.0	50.3
	k=20	50.5	50.8	53.6	49.9	50.1	50.5	51.1	53.5	50.0	50.2	51.1	51.2	54.0	49.8	50.0
labeled	k=1	**60.0**	82.4	82.4	82.3	82.4	66.0	79.0	79.2	79.2	79.2	**57.0**	80.4	80.6	80.6	80.6
	k=3	58.5	86.2	86.2	86.2	86.2	65.0	80.7	81.1	81.1	81.0	56.4	83.8	84.3	84.3	84.3
	k=5	57.3	87.2	87.2	87.2	87.2	65.4	82.7	82.9	82.9	82.8	56.2	84.6	85.0	85.0	85.0
	k=10	57.7	88.9	88.9	88.9	88.9	**66.5**	85.2	85.4	85.4	85.4	56.6	87.0	87.3	87.3	87.3
	k=20	56.4	**89.5**	**89.5**	**89.5**	**89.5**	64.3	85.3	**85.7**	**85.7**	**85.6**	55.4	**87.6**	**87.9**	**87.9**	**88.0**
	k=30	56.3	88.9	88.9	88.9	88.9	63.6	85.4	85.6	85.6	85.6	55.7	87.4	87.6	87.6	87.6

				sw					**te**					**ta**		
		p0	p1	p2	p3	p4	p0	p1	p2	p3	p4	p0	p1	p2	p3	p4
MAJ		50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
Direct		47.3	50.2	51.9	49.9	50.3	50.8	52.5	53.9	49.9	51.4	54.1	59.0	56.2	50.5	51.9
Unlabeled	k=1	**51.4**	**50.4**	**50.5**	**50.5**	**50.1**	**51.6**	**54.8**	**57.5**	**52.3**	**52.1**	**57.1**	**55.3**	**57.2**	**52.6**	**51.6**
	k=3	50.5	50.3	50.3	50.1	50.1	51.3	52.8	55.3	50.6	51.3	55.7	52.5	55.0	50.5	50.6
	k=5	50.6	50.1	50.0	50.1	50.1	51.6	51.7	54.0	50.4	50.3	56.1	51.4	54.0	50.1	50.1
	k=10	50.8	50.1	50.0	50.1	50.1	51.8	52.1	53.5	50.4	50.3	57.3	51.5	53.9	50.0	50.1
	k=20	50.5	50.1	50.0	50.1	50.1	51.4	50.6	52.9	50.0	50.0	56.9	50.5	52.9	50.0	50.0
labeled	k=1	50.5	50.0	49.9	49.9	49.9	**58.2**	75.9	75.8	75.8	75.8	68.1	75.3	75.4	75.4	75.4
	k=3	51.0	54.1	54.1	54.1	54.1	58.0	78.4	78.4	78.4	78.4	70.2	79.1	79.3	79.3	79.2
	k=5	50.7	54.4	54.4	54.4	54.4	56.8	79.1	79.0	79.0	79.1	70.7	80.5	80.5	80.5	80.5
	k=10	**51.3**	**55.5**	**55.5**	**55.5**	**55.5**	57.2	81.3	81.6	81.6	81.6	**70.9**	**83.7**	**83.9**	**83.9**	**83.9**
	k=20	50.9	54.3	54.4	54.4	54.4	56.9	**82.0**	**82.1**	**82.1**	**82.1**	70.8	82.8	83.1	83.1	83.1
	k=30	50.7	54.3	54.3	54.3	54.3	56.8	82.0	82.0	82.0	82.0	70.5	83.3	83.5	83.4	83.4

				mn					**uz**					**my**		
		p0	p1	p2	p3	p4	p0	p1	p2	p3	p4	p0	p1	p2	p3	p4
MAJ		50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
Direct		49.1	49.7	51.4	49.7	50.0	48.5	50.2	52.4	49.7	**51.2**	**54.4**	**56.1**	**56.1**	50.5	**52.6**
Unlabeled	k=1	**51.1**	**54.7**	**58.6**	**52.6**	**52.8**	50.4	**53.1**	**53.6**	**51.8**	50.9	53.0	53.9	56.0	**52.3**	52.0
	k=3	50.2	53.2	56.4	51.0	51.1	50.5	51.9	52.1	50.2	50.3	53.0	51.5	55.0	51.2	50.7
	k=5	50.2	52.0	55.3	50.4	50.5	50.5	50.3	50.7	50.0	50.2	52.9	51.1	53.6	50.5	50.3
	k=10	50.4	52.2	56.3	**50.6**	50.5	50.6	50.3	50.6	50.1	50.0	53.4	51.1	54.2	50.2	50.1
	k=20	50.4	51.1	54.5	50.0	50.0	50.5	50.0	50.7	50.0	50.0	53.2	50.5	52.8	50.0	50.0
labeled	k=1	60.8	74.9	74.9	74.9	74.9	**56.0**	65.0	64.7	64.7	64.7	65.3	73.9	73.8	73.8	73.8
	k=3	60.3	79.5	79.7	79.7	79.7	55.2	65.3	65.2	65.2	65.2	66.6	77.5	77.7	77.7	77.7
	k=5	59.7	80.6	80.6	80.6	80.6	55.5	66.1	66.0	66.0	65.8	65.8	78.6	78.9	78.9	78.9
	k=10	**62.2**	**83.9**	**84.3**	**84.3**	**84.3**	55.9	**68.1**	**68.2**	**68.2**	**68.3**	**67.8**	80.9	81.1	81.1	81.1
	k=20	60.3	82.5	83.2	83.2	83.2	53.8	67.0	67.1	67.1	67.1	67.4	**81.8**	**81.8**	**81.8**	**81.8**
	k=30	59.7	83.3	83.8	83.8	83.8	54.4	67.5	67.7	67.7	67.7	67.6	81.7	81.8	81.8	81.8

				jv					**tl**					**Avg.**		
		p0	p1	p2	p3	p4	p0	p1	p2	p3	p4	p0	p1	p2	p3	p4
MAJ		50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
Direct		**50.9**	52.3	54.1	50.1	**52.3**	49.6	50.4	**51.9**	50.0	**51.2**	50.8	52.5	53.8	50.3	51.4
Unlabeled	k=1	50.6	**53.0**	54.2	**50.9**	50.5	**50.4**	**50.6**	50.9	50.1	50.2	**51.7**	**53.9**	**56.0**	**51.8**	**51.6**
	k=3	50.2	51.7	**53.5**	50.4	50.3	50.0	50.3	50.3	**50.2**	50.0	51.2	52.1	54.4	50.6	50.6
	k=5	50.2	50.9	52.9	50.1	50.2	50.1	50.2	50.1	50.0	50.1	51.4	51.3	53.5	50.2	50.4
	k=10	50.1	50.7	52.5	49.9	50.0	50.2	50.0	50.3	50.0	50.0	51.6	51.3	53.5	50.1	50.2
	k=20	50.5	50.1	51.7	50.0	50.0	50.2	50.0	50.4	50.0	50.0	51.4	50.5	52.5	50.0	50.0
labeled	k=1	**54.1**	59.3	59.3	59.3	59.3	52.4	55.4	55.4	55.4	55.4	58.9	70.1	68.9	70.1	70.1
	k=3	52.7	61.6	61.6	61.6	61.6	52.1	57.7	57.7	57.7	57.7	58.7	73.1	73.2	73.2	73.2
	k=5	52.8	61.5	61.5	61.5	61.5	51.6	60.2	60.2	60.2	60.1	58.4	74.1	74.2	74.2	74.2
	k=10	51.6	**62.6**	**62.6**	**62.6**	**62.6**	**52.4**	**63.2**	**63.3**	**63.3**	**63.3**	**59.1**	**76.4**	**76.5**	**76.5**	**76.5**
	k=20	51.6	61.5	61.5	61.5	61.5	51.5	62.8	62.9	62.9	62.9	58.1	76.1	76.3	76.3	76.3
	k=30	51.6	60.9	61.0	61.0	61.0	51.5	62.3	62.4	62.4	62.4	58.0	76.1	76.2	76.2	76.2

Table 1: Results on Amazon reviews dataset.

pattern 0	[X] [MASK]
pattern 1	[MASK]: [X]
pattern 2	[MASK] News: [X]
pattern 3	[X] Category: [MASK]

		en				**af**				**ur**			
		p0	p1	p2	p3	p0	p1	p2	p3	p0	p1	p2	p3
MAJ		25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
Direct		52.5	47.8	**47.3**	53.0	41.8	41.3	40.2	**57.8**	27.4	32.4	33.0	**53.5**
Unlabeled	k=1	53.7	47.6	45.6	53.2	52.8	**46.8**	**46.2**	53.2	46.2	**41.8**	**41.0**	49.7
	k=3	55.8	47.6	43.4	54.3	53.6	46.5	44.3	54.3	46.2	40.5	38.2	49.9
	k=5	57.1	**48.3**	41.7	**55.6**	54.4	46.9	43.7	55.1	47.0	40.9	37.2	51.4
	k=10	57.5	45.7	41.9	55.3	55.3	44.6	42.3	55.6	46.3	38.3	35.3	51.9
	k=20	**59.7**	46.7	41.5	55.3	**57.2**	45.9	42.2	56.1	**48.1**	39.7	35.5	51.6
labeled	k=1	74.9	83.5	83.8	83.8	75.4	81.2	82.9	82.7	68.1	76.9	78.8	78.7
	k=3	77.1	86.5	86.8	86.7	77.1	84.3	85.4	85.2	69.6	79.4	81.7	81.8
	k=5	78.1	87.7	88.0	87.9	78.6	86.8	87.1	87.1	69.0	79.9	82.7	82.7
	k=10	78.7	88.2	88.5	88.5	79.4	87.2	87.7	87.5	70.5	81.5	**83.6**	**83.4**
	k=20	**79.0**	**89.1**	**89.4**	**89.4**	**79.7**	**87.4**	**87.8**	**87.5**	**70.7**	**81.6**	83.3	83.2
		sw				**te**				**ta**			
		p0	p1	p2	p3	p0	p1	p2	p3	p0	p1	p2	p3
MAJ		25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
Direct		42.5	37.6	33.3	**56.6**	32.2	37.2	32.5	**55.4**	31.3	37.2	28.6	55.1
Unlabeled	k=1	46.5	**42.1**	**42.0**	46.4	46.1	**41.5**	**43.3**	48.6	42.8	**41.6**	**39.2**	47.6
	k=3	**47.1**	41.2	39.9	47.9	**48.2**	40.0	42.4	50.3	44.9	41.0	36.9	50.1
	k=5	47.0	41.5	39.3	48.6	48.0	40.4	41.0	52.4	46.6	39.8	36.0	50.9
	k=10	46.4	38.5	37.0	50.0	47.6	39.0	39.3	51.8	45.6	37.8	33.9	51.5
	k=20	46.7	39.1	36.9	49.9	50.0	40.1	39.7	51.6	**47.9**	38.8	34.7	**52.5**
labeled	k=1	63.5	68.4	70.3	70.3	68.2	73.9	75.0	75.0	64.0	69.7	71.5	71.5
	k=3	65.6	70.8	72.3	72.4	71.1	77.6	78.2	78.2	67.6	74.4	75.7	75.7
	k=5	64.4	72.2	73.5	73.4	**72.9**	79.7	79.9	79.8	68.8	75.8	76.6	76.5
	k=10	67.0	72.5	**74.1**	**73.9**	72.9	79.9	80.0	80.0	68.3	76.5	77.2	77.1
	k=20	**67.5**	**72.7**	73.6	73.6	72.5	**80.2**	**80.6**	**80.6**	**70.0**	**77.5**	**78.1**	**78.2**
		mn				**uz**				**my**			
		p0	p1	p2	p3	p0	p1	p2	p3	p0	p1	p2	p3
MAJ		25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
Direct		31.5	30.9	32.0	47.3	33.0	37.5	33.8	50.7	31.6	37.4	33.7	51.9
Unlabeled	k=1	43.3	**42.5**	**41.5**	48.2	44.3	**44.4**	**42.3**	49.0	45.0	43.9	**43.6**	50.0
	k=3	44.5	41.2	40.5	51.1	46.3	42.2	40.7	50.9	47.1	**44.5**	41.7	53.7
	k=5	44.8	41.5	39.6	51.8	45.8	41.7	39.2	52.3	48.5	43.8	41.4	54.2
	k=10	44.1	39.7	38.0	**53.3**	46.7	39.7	37.9	**53.4**	47.7	41.4	40.0	**54.4**
	k=20	**46.0**	39.7	37.9	52.8	**48.9**	41.2	36.9	53.1	**49.6**	42.2	40.3	53.6
labeled	k=1	62.8	70.9	72.7	72.8	65.6	71.5	73.2	73.3	64.8	76.2	77.4	77.2
	k=3	65.6	75.4	77.3	77.2	68.4	73.6	75.7	75.7	65.9	79.5	80.1	79.8
	k=5	65.9	75.8	78.0	77.9	69.3	76.1	77.9	77.8	66.4	81.4	82.5	81.8
	k=10	66.6	77.0	**78.7**	**78.6**	70.7	76.4	78.3	78.2	67.2	82.4	82.9	82.3
	k=20	**67.5**	**77.4**	78.2	78.0	**70.7**	**77.3**	**78.8**	**78.7**	**68.1**	**83.1**	**83.6**	**83.3**
		jv				**tl**				**Avg**			
		p0	p1	p2	p3	p0	p1	p2	p3	p0	p1	p2	p3
MAJ		25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
Direct		46.9	39.3	38.0	**59.3**	44.8	44.4	42.6	**60.4**	37.8	38.4	36.2	50.9
Unlabeled	k=1	51.0	**45.5**	**45.4**	51.6	49.7	**45.8**	**43.7**	52.2	47.4	**44.2**	**43.5**	48.9
	k=3	52.6	44.6	42.0	53.5	51.0	45.3	42.7	54.0	48.8	43.6	41.9	50.3
	k=5	53.1	44.5	41.3	53.6	52.3	45.2	41.8	54.2	49.5	43.7	41.2	51.0
	k=10	53.0	42.4	39.9	54.0	51.4	44.0	39.8	54.9	49.2	41.7	39.7	51.2
	k=20	**55.4**	42.8	40.1	54.2	**53.2**	44.4	38.9	55.3	**51.1**	42.6	39.9	**51.4**
labeled	k=1	72.5	77.8	79.1	79.1	71.4	76.6	78.9	79.0	68.3	74.6	75.9	75.9
	k=3	74.6	80.5	82.3	82.3	74.4	80.7	82.1	82.2	70.6	77.8	78.9	78.9
	k=5	75.8	81.3	82.8	82.8	75.4	81.2	83.4	83.5	71.3	79.1	80.2	80.1
	k=10	76.6	82.0	84.0	84.2	75.9	82.4	**84.5**	**84.6**	72.1	79.8	80.9	80.8
	k=20	**77.4**	**82.8**	**84.6**	**84.8**	**76.3**	**82.8**	84.0	84.0	**72.6**	**80.4**	**81.1**	**81.1**

Table 2: Results on AG News dataset.

pattern 0	$[X_1]$ [MASK] $[X_2]$
pattern 1	$[X_1]$? [MASK], $[X_2]$ (Yes - No)
pattern 2	$[X_1]$? [MASK], $[X_2]$ (Right - Wrong)

		en			**af**			**ur**			**sw**		
		p0	p1	p2	p0	p1	p2	p0	p1	p2	p0	p1	p2
MAJ		33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3
Direct		33.3	**34.2**	34.3	33.2	33.0	33.4	**33.6**	34.0	33.2	33.2	32.2	33.1
Unlabeled	k=1	**34.1**	33.7	**34.5**	**34.0**	**34.1**	33.7	32.4	**35.3**	32.7	33.5	**33.7**	**33.7**
	k=3	33.7	**34.1**	34.3	33.0	32.9	34.1	33.3	34.0	**33.9**	**33.6**	33.0	33.5
	k=5	31.9	33.7	34.3	32.5	32.8	33.9	31.2	34.1	33.6	33.2	32.7	32.9
	k=10	31.9	33.6	33.3	31.9	33.3	32.6	32.2	34.2	33.2	33.0	32.7	32.5
	k=20	32.0	34.4	33.3	31.6	33.6	34.1	31.6	34.4	33.9	33.1	33.1	32.0
labeled	k=1	38.9	39.1	38.8	38.7	38.9	38.1	37.0	37.4	36.7	33.3	33.4	33.4
	k=3	39.2	39.1	38.6	37.9	37.9	37.4	37.0	37.8	36.8	33.7	33.5	33.7
	k=5	40.0	39.8	39.5	38.0	38.0	37.1	40.2	40.6	39.8	32.7	32.5	32.6
	k=10	41.5	41.6	40.9	41.1	41.1	40.5	42.0	42.4	41.0	33.7	33.7	34.1
	k=20	**44.5**	**44.1**	**43.5**	**42.3**	**43.0**	**41.3**	**42.4**	**43.4**	**42.2**	**35.9**	**35.7**	**35.9**

		te			**ta**			**mn**			**uz**		
		p0	p1	p2	p0	p1	p2	p0	p1	p2	p0	p1	p2
MAJ		33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3
Direct		31.9	33.0	33.2	32.4	34.1	32.9	**33.0**	32.7	32.6	**33.3**	33.3	32.9
Unlabeled	k=1	**34.1**	34.1	**34.1**	**34.5**	34.3	33.3	32.8	33.6	**34.7**	33.2	33.9	32.8
	k=3	32.8	34.9	33.4	33.7	34.7	**34.2**	32.2	**34.5**	33.7	32.3	34.5	33.4
	k=5	32.9	**35.1**	33.8	32.9	34.3	33.9	31.9	33.9	34.1	33.1	**34.5**	**33.9**
	k=10	32.0	34.1	32.7	32.3	34.7	32.5	30.8	34.1	32.5	32.8	33.9	32.6
	k=20	31.5	34.6	32.7	32.5	**34.8**	32.9	32.0	34.1	33.4	32.6	33.5	32.6
labeled	k=1	37.8	38.1	37.7	37.7	38.0	37.0	36.5	36.5	36.5	35.5	34.8	35.0
	k=3	38.9	39.5	38.4	38.7	39.4	37.5	39.1	39.1	38.9	35.1	34.7	34.7
	k=5	37.5	37.1	35.9	38.3	38.7	36.3	37.1	36.9	36.9	36.0	35.9	35.9
	k=10	39.2	39.5	37.9	41.1	40.8	38.0	39.5	39.3	39.3	38.3	37.9	37.8
	k=20	**41.2**	**41.5**	**39.3**	**42.7**	**43.1**	**39.7**	**40.3**	**40.2**	**40.0**	**40.0**	**39.9**	**39.6**

		my			**jv**			**tl**			**Avg**		
		p0	p1	p2	p0	p1	p2	p0	p1	p2	p0	p1	p2
MAJ		33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3
Direct		**33.7**	33.6	33.7	**33.3**	33.3	33.6	33.3	33.5	32.3	33.1	33.3	33.1
Unlabeled	k=1	33.3	33.5	**33.8**	32.4	32.0	33.3	33.8	32.7	32.8	**33.4**	33.7	33.5
	k=3	32.6	33.9	33.7	32.1	31.4	34.2	33.7	**33.9**	**33.3**	32.9	**33.7**	**33.7**
	k=5	32.5	**34.3**	33.6	32.4	31.6	34.3	**34.1**	33.5	32.1	32.7	33.6	33.6
	k=10	30.5	33.9	33.3	32.1	32.6	33.5	33.2	33.1	32.6	32.1	33.5	32.8
	k=20	30.9	33.5	32.7	30.8	**33.6**	**34.7**	32.9	32.5	33.1	32.0	33.6	33.2
labeled	k=1	36.8	36.7	36.1	34.2	33.5	33.3	34.7	34.4	34.3	36.2	36.2	35.8
	k=3	36.7	36.9	36.2	34.6	33.9	33.9	35.7	35.7	35.7	36.7	36.8	36.3
	k=5	37.7	37.7	37.3	**35.2**	**34.8**	**34.6**	35.7	35.7	35.3	36.9	36.8	36.2
	k=10	39.5	39.3	38.1	34.7	34.4	33.6	37.2	36.9	36.9	38.6	38.5	37.7
	k=20	**41.7**	**41.3**	**39.6**	32.8	32.8	32.4	**37.4**	**37.0**	**37.0**	**39.7**	**39.8**	**38.7**

Table 3: Results on XNLI dataset.

B Experimental Details of Decomposed Prompting Work

B.1 Prompt Details

Zero- and few-shot prompts used in this work are shown in Figure 1 (decomposed prompting) and Figure 2 (iterative prompting).

B.2 Full Results

Full experimental results are displayed in Table 4 (Mistral 7B), Table 5 (LLaMA2 7B), Table 6 (LLaMA 13B), Table 7 (BLOOMZ 7B), and Table 8 (mTk 13B).

	language	en	af	ar	bg	de	el	es	et	eu	fa	fi	fr	he
zero-shot	Iter	65.2	67.8	57.2	68.6	65.0	55.0	64.8	49.4	35.6	58.3	50.2	65.4	51.5
	Decom (prob.)	63.6	66.0	67.8	74.4	68.6	62.7	68.6	58.0	54.1	68.5	60.2	63.5	66.4
	Decom (gen.)	45.3	43.8	49.6	50.5	49.0	50.7	43.3	53.6	50.7	56.0	55.5	40.5	55.6
few-shot	Iter	80.2	66.4	65.0	77.3	66.9	56.4	70.8	53.7	50.7	57.4	63.9	67.7	66.4
	Decom (prob.)	85.0	76.9	48.1	82.4	78.3	52.3	82.7	65.2	48.8	57.3	64.4	76.9	66.6
	Decom (gen.)	81.4	74.8	44.3	80.4	77.0	46.3	82.0	64.0	48.1	54.1	63.6	76.4	64.9
	Decom (prob.) + I	83.4	77.9	42.4	76.9	77.8	33.6	77.6	64.6	57.4	42.9	67.6	74.8	58.5
	Decom (gen.) + I	78.7	75.8	34.0	74.9	76.6	24.7	76.4	62.6	56.8	34.4	64.5	73.4	54.5
	language	hi	hu	id	it	ja	kk	ko	lt	mr	nl	pl	pt	ro
zero-shot	Iter	61.3	50.6	54.7	64.0	42.2	36.7	39.9	52.8	39.1	60.4	66.5	63.9	66.2
	Decom (prob.)	37.1	58.6	61.0	68.6	56.3	57.8	47.4	68.2	61.0	69.4	73.5	68.4	68.5
	Decom (gen.)	35.6	46.7	41.8	45.1	48.9	50.2	42.2	60.3	56.7	46.8	59.5	43.1	44.6
few-shot	Iter	65.7	50.4	70.0	67.2	42.0	43.8	42.6	63.2	54.4	66.6	70.9	75.1	65.9
	Decom (prob.)	67.8	71.3	73.9	76.2	59.8	50.0	44.0	67.5	48.9	80.6	78.6	77.8	77.8
	Decom (gen.)	66.2	70.8	73.0	76.0	57.1	50.2	43.4	67.1	48.9	77.2	78.3	76.9	77.0
	Decom (prob.) + I	57.6	66.5	70.4	72.2	54.2	58.4	49.2	69.9	53.1	78.5	76.7	75.0	76.4
	Decom (gen.) + I	55.3	63.9	68.2	70.3	53.1	57.9	48.2	69.5	52.7	76.9	75.7	74.2	75.1
	language	ru	ta	te	th	tl	tr	uk	ur	vi	wo	yo	zh	avg.
zero-shot	Iter	68.2	39.2	51.1	54.1	65.0	47.7	67.0	56.0	41.7	31.5	41.3	58.8	54.3
	Decom (prob.)	74.4	55.2	63.8	63.0	62.9	55.2	74.1	54.2	59.9	39.6	49.7	59.2	61.8
	Decom (gen.)	54.7	52.2	57.4	50.1	51.3	43.2	57.4	40.3	45.9	29.2	43.3	55.7	48.7
few-shot	Iter	74.0	52.0	62.4	57.1	37.3	62.0	68.2	59.6	41.0	25.2	39.0	62.3	58.9
	Decom (prob.)	79.9	37.5	61.4	58.2	73.4	62.7	77.7	51.3	52.6	42.0	47.8	65.8	64.4
	Decom (gen.)	78.0	33.9	61.3	56.9	73.4	62.6	76.2	45.7	52.8	42.0	47.6	64.5	63.0
	Decom (prob.) + I	76.8	35.7	67.0	45.8	74.9	63.7	75.1	40.5	59.4	43.1	49.2	62.9	62.3
	Decom (gen.) + I	73.9	28.0	66.6	42.9	74.9	62.6	73.4	32.9	59.7	43.2	48.6	61.4	59.9

Table 4: Full results on Mistral 7b.

	language	en	af	ar	bg	de	el	es	et	eu	fa	fi	fr	he
zero-shot	Iter	33.1	38.8	30.2	33.2	34.5	38.1	38.9	19.7	11.8	17.7	26.0	37.5	21.3
	Decom (prob.)	58.2	45.1	49.6	55.9	53.3	50.4	44.7	37.7	36.4	40.5	41.3	46.8	39.5
	Decom (gen.)	53.8	46.8	38.5	45.8	57.1	54.3	52.4	28.6	20.2	35.9	39.8	53.1	37.5
few-shot	Iter	68.0	56.1	58.0	63.4	56.9	48.7	55.3	46.5	41.3	51.1	50.5	54.2	54.0
	Decom (prob.)	74.7	60.0	29.9	64.7	63.0	30.6	55.7	53.0	44.4	29.7	62.9	54.4	42.8
	Decom (gen.)	62.1	51.0	25.7	60.3	52.4	23.9	50.3	48.3	42.9	26.0	56.8	49.5	37.5
	Decom (prob.) + I	68.2	55.9	23.7	61.6	61.0	20.2	52.5	43.2	40.8	22.7	49.4	54.8	35.4
	Decom (gen.) + I	63.4	53.2	19.0	57.9	56.2	12.0	47.8	39.3	40.0	15.5	46.4	51.2	30.1
	language	hi	hu	id	it	ja	kk	ko	lt	mr	nl	pl	pt	ro
zero-shot	Iter	35.2	29.3	31.1	35.1	28.7	13.6	19.8	24.9	13.2	37.5	37.7	38.4	32.0
	Decom (prob.)	36.9	47.0	46.9	46.7	32.4	39.0	29.0	34.9	45.3	54.9	54.0	48.6	43.6
	Decom (gen.)	34.8	47.4	39.1	45.2	30.9	33.0	33.2	37.7	42.0	51.1	44.1	48.5	42.6
few-shot	Iter	54.0	41.0	51.3	49.6	40.0	43.2	25.0	52.5	50.3	52.2	52.4	52.0	53.8
	Decom (prob.)	45.8	62.6	60.9	56.4	40.2	51.4	48.2	56.3	47.3	58.9	67.2	60.3	63.6
	Decom (gen.)	42.4	57.0	56.5	51.6	34.1	47.5	44.7	51.7	43.5	51.3	64.2	54.5	55.5
	Decom (prob.) + I	30.6	52.3	54.1	51.3	37.3	46.6	41.9	46.5	45.7	64.2	65.4	55.2	56.4
	Decom (gen.) + I	24.1	50.6	49.5	44.1	32.9	46.0	40.7	45.3	34.5	60.2	62.0	51.2	51.8
	language	ru	ta	te	th	tl	tr	uk	ur	vi	wo	yo	zh	avg.
zero-shot	Iter	29.8	19.2	13.8	29.2	28.6	22.2	30.3	20.7	29.7	13.3	13.7	32.2	27.2
	Decom (prob.)	55.8	38.0	34.0	37.5	57.3	48.3	57.4	31.6	39.5	27.6	29.1	42.9	43.2
	Decom (gen.)	48.7	25.5	36.9	34.6	66.3	45.9	48.8	28.4	35.3	18.7	21.8	44.0	40.4
few-shot	Iter	58.2	30.9	54.3	49.4	37.3	34.4	57.7	44.0	46.5	40.7	39.3	52.0	48.6
	Decom (prob.)	67.2	31.7	44.7	36.5	46.8	58.1	62.9	27.1	41.4	39.9	37.1	64.8	50.5
	Decom (gen.)	62.3	25.3	43.5	34.7	45.4	55.9	59.4	23.7	40.7	36.2	35.5	50.9	45.8
	Decom (prob.) + I	59.6	20.3	38.4	20.9	63.1	54.1	59.9	19.3	49.7	32.2	33.8	48.2	45.1
	Decom (gen.) + I	56.9	12.5	34.5	16.7	58.8	52.7	57.5	13.0	47.8	29.7	31.7	44.2	41.0

Table 5: Full results on LLaMA2 7b.

Zero-shot prompt

```
POS tag set:  ADJ ADP ADV AUX CCONJ DET INTJ NOUN NUM PART PRON PROPN
PUNCT SCONJ SYM VERB X
Sentence:  Viel Erfolg !
In the sentence, the part-of-speech tag of ‘Viel’ is a kind of
```

Few-shot prompt (w/o Instruction)

```
Sentence:  And if you send me a story , that would be great !
In the sentence, the part-of-speech tag of ‘if’ is a kind of SCONJ.
Sentence:  I ‘ll admit I was n’t expecting much from this place , but they
really did do a good job .
In the sentence, the part-of-speech tag of ‘good’ is a kind of ADJ.
Sentence:  I do n’t know .  The girl shrugged once again .  In the
sentence, the part-of-speech tag of ‘girl’ is a kind of NOUN.
Sentence:  The dancers were falling back round a Polish agriculturalist
who was teaching a gangling Englishman and two young Africans an Eastern
European peasant dance .
In the sentence, the part-of-speech tag of ‘around’ is a kind of ADP.
Sentence:  Antigua was awesome .
In the sentence, the part-of-speech tag of ‘was’ is a kind of AUX.
Sentence:  The food is fresh and taste great .
In the sentence, the part-of-speech tag of ‘the’ is a kind of DET.
Sentence:  Now I have wife and son .
In the sentence, the part-of-speech tag of ‘Now’ is a kind of ADV.
Sentence:  However , this fruitful period was short-lived , as Greece
suffered badly under the Ottoman Empire , only to recover in the 19th
century as the capital of independent Greece .
In the sentence, the part-of-speech tag of ‘suffered’ is a kind of VERB.
Sentence:  I survived it without a problem .
In the sentence, the part-of-speech tag of ‘.’  is a kind of PUNCT.
Sentence:  The food is fresh and taste great .
In the sentence, the part-of-speech tag of ‘and’ is a kind of CCONJ.
Sentence:  you can view at dresscod.com
In the sentence, the part-of-speech tag of ‘dresscod.com’ is a kind of X.
Sentence:  I do n’t know .  The girl shrugged once again .
In the sentence, the part-of-speech tag of ‘I’ is a kind of PRON.
Sentence:  I ‘ll admit I was n’t expecting much from this place , but they
really did do a good job .
In the sentence, the part-of-speech tag of ‘n’t’ is a kind of PART.
Sentence:  Antigua was awesome .
In the sentence, the part-of-speech tag of ‘Antigua’ is a kind of PROPN.
Sentence:  The dancers were falling back round a Polish agriculturalist
who was teaching a gangling Englishman and two young Africans an Eastern
European peasant dance .
In the sentence, the part-of-speech tag of ‘two’ is a kind of NUM.
Sentence:  Yes , the Cyclone is almost certain to lose strength as it
surges over land .
In the sentence, the part-of-speech tag of ‘Yes’ is a kind of INTJ.
Sentence:  ----== Posted via Newsfeed.Com - Unlimited - Uncensored -
Secure Usenet News ==----
In the sentence, the part-of-speech tag of ‘----== ‘ is a kind of SYM.
Sentence:  Viel Erfolg !  In the sentence, the part-of-speech tag of
‘Viel’ is a kind of
```

Figure 1: Prompt design of decomposed prompting.

Zero-shot prompt

```
POS tag set: ADJ ADP ADV AUX CCONJ DET INTJ NOUN NUM PART PRON PROPN
PUNCT SCONJ SYM VERB X
Sentence: Viel Erfolg !
Viel_
```

Few-shot prompt (w/o Instruction)

```
Context: Chahine said her immediate family spent about $ 20,000 to return
to Detroit via Syria and Jordan .
Tagged: Chahine_PROPN said_VERB her_PRON immediate_ADJ family_NOUN
spent_VERB about_ADV $_SYM 20,000_NUM to_PART return_VERB to_ADP Detroit_PROPN
via_ADP Syria_PROPN and_CCONJ Jordan_PROPN ._PUNCT
Context: Welcome Darin !
Tagged: Welcome_INTJ Darin_PROPN !_PUNCT
Context: you can view at dresscod.com
Tagged: you_PRON can_AUX view_VERB at_ADP dresscod.com_X
...
Context: They work on Wall Street , after all , so when they hear a
company who's stated goals include " Do n't be evil , " they imagine a
company who's eventually history will be " Do n't be profitable . "
Tagged: They_PRON work_VERB on_ADP Wall_PROPN Street_PROPN ,_PUNCT after_ADV
all_ADV ,_PUNCT so_ADV when_ADV they_PRON hear_VERB a_DET company_NOUN
who's_PRON stated_VERB goals_NOUN include_VERB "_PUNCT Do_AUX n't_PART be_AUX
evil_ADJ ,_PUNCT "_PUNCT they_PRON imagine_VERB a_DET company_NOUN who's_PRON
eventually_ADJ history_NOUN will_AUX be_VERB "_PUNCT Do_AUX n't_PART be_AUX
profitable_ADJ ._PUNCT "_PUNCT
Context: It 's not quite as freewheeling an environment as you 'd imagine
: Sergey Brin has actually created a mathematical ' proof ' that the
company 's self - driven research strategy , which gives employees one day
a week to do research projects on their own , is a good , respectable idea
.
Tagged: It_PRON 's_AUX not_PART quite_ADV as_ADV freewheeling_ADJ an_DET
environment_NOUN as_SCONJ you_PRON 'd_AUX imagine_VERB :_PUNCT Sergey_PROPN
Brin_PROPN has_AUX actually_ADV created_VERB a_DET mathematical_ADJ '_PUNCT
proof_NOUN '_PUNCT that_SCONJ the_DET company_NOUN 's_PART self_NOUN -_PUNCT
driven_VERB research_NOUN strategy_NOUN ,_PUNCT which_PRON gives_VERB
employees_NOUN one_NUM day_NOUN a_DET week_NOUN to_PART do_VERB research_NOUN
projects_NOUN on_ADP their_PRON own_ADJ ,_PUNCT is_AUX a_DET good_ADJ ,_PUNCT
respectable_ADJ idea_NOUN ._PUNCT
Context: Read the entire article ; there 's a punchline , too .
Tagged: Read_VERB the_DET entire_ADJ article_NOUN ;_PUNCT there_PRON 's_VERB
a_DET punchline_NOUN ,_PUNCT too_ADV ._PUNCT
Context: My opinion piece on the implications of Arafat 's passing for al
- Qaeda has appeared at Newsday .
Tagged: My_PRON opinion_NOUN piece_NOUN on_ADP the_DET implications_NOUN
of_ADP Arafat_PROPN 's_PART passing_NOUN for_ADP al_PROPN -_PUNCT Qaeda_PROPN
has_AUX appeared_VERB at_ADP Newsday_PROPN ._PUNCT
Context: Viel Erfolg ! Tagged: Viel_
```

Figure 2: Prompt design of iterative prompting.

	language	en	af	ar	bg	de	el	es	et	eu	fa	fi	fr	he
zero-shot	Iter	47.6	37.4	43.2	44.5	45.7	38.4	46.8	37.0	26.5	42.0	40.7	45.5	40.0
	Decom (prob.)	67.3	60.1	54.4	62.7	63.6	60.5	55.9	49.9	37.4	59.8	62.6	53.4	55.4
	Decom (gen.)	59.2	54.1	45.0	52.5	57.5	51.3	56.3	37.6	36.7	49.7	50.2	54.7	44.3
few-shot	Iter	68.0	62.3	57.4	69.9	60.3	57.9	66.7	44.8	41.0	49.1	54.2	63.2	59.8
	Decom (prob.)	77.3	67.8	33.2	67.6	67.5	35.0	62.6	58.5	46.9	34.7	62.8	64.8	48.4
	Decom (gen.)	65.3	59.1	25.1	61.3	58.6	24.6	53.5	51.8	45.8	27.4	55.4	55.9	43.9
	Decom (prob.) + I	74.3	67.6	25.9	60.7	70.5	21.5	59.1	51.4	44.1	21.8	59.1	63.1	40.3
	Decom (gen.) + I	68.7	64.4	19.2	58.7	66.2	12.4	53.9	47.9	42.2	15.5	54.0	59.7	35.0
	language	hi	hu	id	it	ja	kk	ko	lt	mr	nl	pl	pt	ro
zero-shot	Iter	45.0	38.8	40.9	41.8	42.8	24.1	29.8	41.2	30.5	36.6	42.2	43.3	43.1
	Decom (prob.)	53.8	57.6	57.4	54.8	48.3	51.8	45.1	54.3	50.2	62.0	66.4	56.6	57.9
	Decom (gen.)	45.4	47.9	48.2	51.3	35.9	48.7	35.3	43.2	48.7	56.9	58.2	51.3	51.4
few-shot	Iter	51.6	46.1	60.8	62.7	46.5	32.0	26.6	50.8	52.7	61.0	64.4	68.9	58.9
	Decom (prob.)	45.4	69.8	62.2	61.2	44.6	52.3	46.1	63.0	49.6	65.4	68.1	62.3	63.6
	Decom (gen.)	37.3	60.5	55.8	54.5	40.7	49.4	42.6	58.4	46.9	54.9	61.4	54.3	54.9
	Decom (prob.) + I	31.4	64.2	55.3	55.3	38.1	51.7	47.1	58.9	52.5	65.4	60.2	56.3	60.4
	Decom (gen.) + I	23.4	60.0	50.2	52.4	35.5	49.0	45.3	56.9	50.8	61.1	58.2	54.1	56.1
	language	ru	ta	te	th	tl	tr	uk	ur	vi	wo	yo	zh	avg.
zero-shot	Iter	42.6	21.8	22.5	45.6	29.3	29.9	39.8	35.1	36.0	24.4	24.1	45.2	37.4
	Decom (prob.)	66.5	49.1	50.8	44.6	66.5	56.9	65.7	47.2	45.3	34.5	47.7	58.7	54.7
	Decom (gen.)	55.2	46.2	54.1	44.2	73.1	52.8	57.3	40.2	45.4	29.9	39.6	52.5	48.7
few-shot	Iter	64.9	33.5	51.5	51.5	60.2	46.3	61.6	45.4	41.8	36.3	31.6	52.1	52.6
	Decom (prob.)	71.0	30.4	54.4	40.1	74.0	54.1	69.0	30.1	47.5	39.4	36.2	66.6	54.5
	Decom (gen.)	63.3	21.9	51.3	33.9	70.9	52.2	61.4	22.1	45.2	38.1	34.8	56.5	48.3
	Decom (prob.) + I	63.3	22.3	52.2	23.5	70.7	53.9	62.4	19.0	48.4	36.9	36.4	56.7	49.4
	Decom (gen.) + I	59.8	14.1	48.4	18.5	70.2	53.2	59.1	12.0	47.1	34.5	34.5	52.7	45.6

Table 6: Full results on LLaMA2 13b.

	language	en	af	ar	bg	de	el	es	et	eu	fa	fi	fr	he
zero-shot	Iter	6.4	7.2	10.9	7.6	9.5	8.4	8.2	12.4	7.5	7.3	9.3	9.0	9.6
	Decom (prob.)	20.6	20.5	14.5	19.7	26.2	18.3	18.2	22.3	19.0	12.8	19.2	19.4	15.2
	Decom (gen.)	28.7	18.3	16.4	22.6	26.8	22.7	24.9	21.2	25.0	11.3	20.9	20.9	21.8
few-shot	Iter	30.9	6.4	14.4	23.8	19.3	7.7	23.2	16.6	28.4	11.1	22.3	25.1	7.5
	Decom (prob.)	44.1	33.1	28.7	35.9	44.0	39.2	33.6	39.0	38.4	25.6	38.5	35.6	34.3
	Decom (gen.)	40.6	31.0	25.5	31.4	39.5	35.8	30.5	36.9	33.8	21.6	36.8	31.0	33.6
	Decom (prob.) + I	33.3	24.7	27.2	35.2	30.0	31.0	30.1	36.5	37.4	24.7	34.4	29.0	29.2
	Decom (gen.) + I	33.3	24.5	27.1	35.0	29.7	30.4	30.0	36.4	37.1	24.5	34.5	28.9	29.1
	language	hi	hu	id	it	ja	kk	ko	lt	mr	nl	pl	pt	ro
zero-shot	Iter	3.9	13.0	10.0	9.1	2.8	4.5	8.5	7.8	0.4	9.1	9.9	8.6	8.8
	Decom (prob.)	12.0	27.0	17.7	23.1	13.5	17.7	19.5	23.6	12.4	18.6	23.6	19.5	19.6
	Decom (gen.)	15.2	21.9	17.3	26.2	26.2	16.8	21.3	23.4	25.8	14.7	23.2	27.8	24.3
few-shot	Iter	20.5	13.4	30.5	19.0	6.3	17.0	5.9	15.0	35.2	20.8	17.9	27.4	13.4
	Decom (prob.)	27.0	38.2	43.8	33.9	25.9	45.6	35.0	40.3	39.6	39.8	39.7	34.4	33.3
	Decom (gen.)	24.8	36.9	41.2	31.1	22.5	43.8	32.7	39.5	28.0	36.5	36.5	31.7	32.0
	Decom (prob.) + I	25.6	32.3	36.0	30.7	25.3	45.2	27.7	41.0	44.5	29.0	34.7	30.4	32.5
	Decom (gen.) + I	25.6	32.2	35.9	30.6	25.1	45.1	27.7	41.0	43.7	28.6	34.6	30.3	32.5
	language	ru	ta	te	th	tl	tr	uk	ur	vi	wo	yo	zh	avg.
zero-shot	Iter	6.8	5.0	5.1	6.8	3.9	9.0	5.2	6.6	4.2	1.4	7.2	7.6	7.4
	Decom (prob.)	26.1	15.0	7.9	8.7	7.8	15.5	23.7	8.1	14.4	11.0	18.9	21.7	17.6
	Decom (gen.)	27.9	20.7	12.8	2.7	1.9	17.4	28.1	12.8	25.7	21.1	28.3	26.0	20.6
few-shot	Iter	20.3	24.3	47.0	3.1	22.5	20.9	20.9	15.5	18.3	16.5	16.9	20.7	18.8
	Decom (prob.)	41.9	36.5	48.2	25.0	41.9	37.9	39.6	26.2	26.9	34.1	39.2	40.8	36.2
	Decom (gen.)	36.8	33.5	41.7	23.1	41.9	36.4	37.0	24.7	24.5	33.2	36.5	35.7	33.2
	Decom (prob.) + I	37.0	34.1	39.0	13.7	57.8	38.0	35.8	26.4	34.0	30.3	33.3	32.8	32.9
	Decom (gen.) + I	36.9	33.9	38.8	13.6	57.8	38.0	35.4	26.4	33.9	30.3	33.3	32.6	32.7

Table 7: Full results on BLOOMZ 7b.

	language	en	af	ar	bg	de	el	es	et	eu	fa	fi	fr	he
zero-shot	Decom (gen.)	47.6	45.7	37.8	48.9	48.9	45.8	40.0	45.3	41.5	44.2	46.8	42.6	42.6
few-shot	Decom (gen.)	49.0	41.0	16.2	37.6	43.9	31.0	37.2	34.8	33.9	33.4	32.1	38.5	34.1
	Decom (gen.) + I	57.3	51.9	27.4	47.2	55.4	40.1	50.1	41.2	43.6	48.1	42.4	49.9	45.6
	language	hi	hu	id	it	ja	kk	ko	lt	mr	nl	pl	pt	ro
zero-shot	Decom (gen.)	40.6	38.7	39.3	39.3	32.9	46.1	29.2	47.4	47.5	42.8	46.1	40.6	49.4
few-shot	Decom (gen.)	23.8	33.5	39.9	36.5	14.3	32.4	17.7	37.5	34.9	42.7	36.1	37.1	35.6
	Decom (gen.) + I	44.7	36.2	51.9	45.7	44.6	45.7	26.7	45.7	48.8	55.3	46.2	48.9	51.5
	language	ru	ta	te	th	tl	tr	uk	ur	vi	wo	yo	zh	avg.
zero-shot	Decom (gen.)	45.9	39.4	51.3	47.1	59.3	46.9	47.4	37.9	48.4	22.3	37.5	42.8	43.1
few-shot	Decom (gen.)	33.5	28.1	50.9	21.9	65.7	34.7	31.2	17.7	33.9	10.5	22.4	17.2	32.5
	Decom (gen.) + I	43.8	38.0	55.3	46.6	70.5	46.0	41.5	36.0	49.0	19.8	38.6	34.5	44.7

Table 8: Full results on mTk 13b.

C BMIKE-53 Details

C.1 Data Entry Example

Figure 3 shows the data item examples of BMIKE-53 for all three datasets.

C.2 Full Results

We show the full experimental results for all three tasks in Table 9 (zsRE), Table 10 (CounterFact), and Table 11 (WFD), respectively.

zsRe

Llama3.2-3B

0-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	53.2	32.6	62.0	36.1	44.8	49.4	46.1	68.9	60.4	27.6	63.8	44.5	31.4	39.4	57.0	58.3	29.0	69.2	47.2	41.8	45.8	46.6	48.5	56.4	63.3	45.3	74.0	43.9	37.1	49.5	34.0	55.9	53.0	53.4	65.8	44.9	59.4	41.2	59.2	50.3	62.0	49.0	59.5	45.3	67.7	49.2	38.4	59.8	56.4	36.6	57.6	36.9	94.3	50.2
rel_em	44.7	22.5	55.3	30.0	28.7	15.1	37.6	61.2	54.2	20.0	59.0	38.2	20.7	32.4	49.7	49.7	20.5	63.9	38.9	35.4	37.0	19.2	33.1	49.7	53.8	17.6	69.7	37.8	23.3	17.0	19.0	46.0	44.9	45.2	59.0	38.1	52.8	33.9	51.4	35.9	55.8	43.5	52.6	36.0	63.0	15.5	27.5	53.0	43.2	24.6	50.3	28.4	93.1	39.5
gen_f1	47.6	33.5	59.1	35.4	41.8	48.7	44.1	67.3	59.2	37.6	62.5	45.7	31.9	34.0	53.9	56.7	28.1	65.8	45.4	41.4	45.7	45.0	47.5	54.7	61.2	44.7	72.2	44.6	35.9	48.7	32.9	54.7	52.8	52.2	64.5	44.9	59.3	40.1	58.0	49.7	59.9	46.6	56.9	44.6	66.0	49.1	38.5	56.7	55.5	35.1	55.2	36.7	87.3	49.0
gen_em	39.8	22.8	52.1	30.0	26.1	14.4	35.3	59.1	53.0	30.0	57.5	39.2	21.8	27.9	46.8	47.9	19.8	60.7	38.1	34.9	36.1	18.2	32.8	48.7	52.2	16.0	67.1	38.2	22.2	15.9	18.8	44.5	44.3	44.1	57.6	39.0	52.6	33.0	50.1	35.4	54.0	41.2	49.9	35.2	61.6	14.8	28.8	49.9	42.4	23.5	48.0	27.1	84.1	38.5
loc_f1	4.3	2.5	4.3	14.4	3.2	16.5	4.3	3.6	4.3	0.2	6.0	5.6	2.6	4.8	3.3	2.7	1.8	4.7	5.6	1.4	4.5	10.1	5.2	4.1	4.6	18.8	7.2	5.9	5.1	24.4	5.6	2.5	3.1	2.9	5.2	6.3	5.1	5.8	4.5	3.8	3.6	2.4	1.7	3.0	6.3	19.1	4.0	5.2	3.6	3.8	5.0	4.9	12.9	5.6
loc_em	1.8	0.5	1.9	10.0	0.9	0.3	1.9	1.5	1.6	0.0	3.2	2.7	0.5	3.0	1.8	1.1	0.1	2.7	2.7	0.1	1.9	0.1	1.9	2.3	1.5	0.4	4.5	3.5	0.4	0.4	1.2	1.2	1.1	0.7	3.0	3.0	2.3	3.0	2.6	0.8	1.6	1.4	0.3	1.6	3.5	0.7	1.4	2.6	1.2	0.4	2.2	0.7	7.3	1.8
port_f1	3.5	3.4	5.5	0.0	4.0	19.2	3.4	4.1	4.2	1.8	3.5	2.9	3.1	3.6	3.0	3.6	4.0	3.7	4.1	3.0	4.7	14.6	5.5	3.4	5.7	19.9	6.0	3.3	4.3	24.7	3.1	3.4	3.3	3.1	4.5	3.9	4.2	3.5	4.0	6.4	4.2	2.7	3.4	2.5	5.2	22.6	3.4	4.6	4.6	3.9	4.5	5.0	24.1	5.4
port_em	0.8	0.5	2.6	0.0	0.7	0.3	0.9	1.4	1.5	0.0	0.9	0.8	0.1	0.9	0.4	1.1	0.9	1.1	1.2	1.1	1.6	0.7	1.8	1.2	2.6	0.7	2.4	1.2	0.4	0.4	0.7	0.9	0.7	0.4	1.4	1.1	1.5	1.2	1.8	1.9	1.5	1.1	0.7	0.3	2.2	0.4	0.5	1.6	1.5	0.4	2.0	1.6	17.1	1.1

1-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	76.5	57.5	69.6	62.6	65.1	57.7	78.5	85.1	80.9	73.2	82.5	85.4	52.3	80.0	77.0	71.2	50.7	78.7	81.2	68.7	73.2	64.9	69.6	75.5	77.5	58.4	84.6	80.3	72.4	59.4	62.4	65.1	69.5	72.4	79.5	81.4	79.1	77.9	81.8	71.0	78.3	76.9	73.0	60.4	83.1	53.4	59.8	77.8	71.0	56.0	76.4	65.7	97.1	71.6
rel_em	70.3	41.9	58.8	46.3	48.3	24.2	70.8	79.1	75.2	67.7	77.5	81.3	40.0	73.2	69.9	62.6	37.4	73.2	72.3	64.1	64.9	35.7	51.5	69.0	70.4	25.8	80.2	73.4	58.1	21.6	45.9	56.1	60.6	64.6	72.9	77.0	73.1	71.6	73.4	57.9	72.6	70.8	61.5	51.2	77.7	17.7	46.8	70.6	59.1	36.7	70.4	53.6	96.4	60.1
gen_f1	77.0	57.7	68.5	62.6	65.6	56.9	77.4	85.0	81.0	73.3	81.8	85.2	51.3	78.9	76.4	71.3	50.1	78.6	80.3	68.2	72.6	64.7	69.6	75.4	77.2	58.1	85.2	80.1	71.9	59.7	62.4	65.1	69.5	72.4	78.8	81.1	77.9	78.4	81.5	70.4	78.6	76.7	72.6	59.8	82.5	53.4	59.2	77.0	71.0	54.9	74.8	65.4	96.8	71.2
gen_em	71.1	42.4	57.5	46.2	49.0	23.8	69.7	78.6	75.6	68.1	76.5	81.2	39.2	72.1	68.8	62.9	36.7	72.8	71.1	63.5	63.3	35.4	51.1	68.8	69.3	25.4	80.9	73.6	57.6	21.6	46.2	56.0	60.5	64.3	72.2	76.6	71.8	72.4	73.6	57.3	72.8	70.9	61.4	50.5	77.4	17.9	46.2	69.9	58.9	35.8	68.7	52.8	96.0	59.8
loc_f1	9.1	3.9	4.5	4.0	4.9	17.6	8.6	4.4	9.1	2.1	10.0	10.6	4.6	9.9	6.7	1.7	3.3	7.0	9.9	3.5	8.2	11.6	5.1	8.7	7.0	19.0	8.0	9.0	6.1	24.2	5.5	2.7	5.5	4.8	8.2	10.9	8.5	12.1	9.7	6.0	7.2	6.5	5.3	4.4	9.7	19.7	4.4	6.4	4.8	4.7	8.3	6.9	18.6	7.8
loc_em	5.7	0.9	1.9	1.6	2.3	0.9	4.7	2.4	5.7	1.2	6.2	7.3	1.6	5.4	4.4	0.7	1.2	4.3	5.3	1.6	5.3	0.7	2.0	5.8	3.8	0.4	5.0	5.4	0.9	0.8	1.4	1.6	2.2	2.3	5.1	6.6	5.3	7.9	6.5	2.8	4.2	4.2	2.7	1.9	6.4	0.7	1.5	3.9	2.7	0.9	3.8	1.9	12.8	3.3
port_f1	17.0	9.5	8.8	11.0	14.5	19.7	22.0	12.2	20.6	7.1	19.2	21.9	11.3	20.3	11.5	6.0	10.3	15.9	23.4	7.6	20.3	19.2	10.7	16.3	15.3	23.7	12.6	24.6	11.7	27.5	7.5	8.2	8.6	7.9	14.7	20.4	17.9	22.1	17.5	16.5	14.0	15.2	11.5	8.4	18.3	23.9	7.9	15.4	15.1	7.3	13.4	15.3	38.6	15.0
port_em	9.8	4.6	4.0	5.1	6.7	1.1	14.9	7.9	14.3	3.1	13.2	14.9	4.6	14.3	6.1	2.4	4.0	9.3	15.3	4.0	13.7	2.4	4.5	10.4	8.9	1.2	7.8	17.4	3.8	0.9	3.0	4.1	4.3	3.8	9.2	13.2	12.6	14.9	11.2	10.0	8.8	9.4	6.8	4.3	12.8	0.5	2.6	10.0	8.5	1.5	8.8	7.7	29.9	7.7

8-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	74.6	57.5	68.5	60.3	64.1	56.0	80.3	83.7	80.0	75.7	82.4	85.1	51.8	76.6	77.4	74.0	52.3	77.2	80.4	68.8	79.9	68.7	66.8	76.6	77.0	55.3	82.8	78.7	74.2	54.8	58.7	63.1	70.8	65.5	77.9	80.7	80.2	75.8	80.5	67.8	77.8	77.5	71.9	36.7	84.1	60.3	54.1	79.0	62.8	55.4	75.5	71.0	99.3	70.5
rel_em	68.0	42.4	58.8	47.1	49.7	20.2	72.3	77.9	75.1	69.3	76.7	81.3	42.5	69.2	69.3	64.6	39.3	70.9	70.7	63.3	72.3	41.3	47.3	70.0	68.5	21.3	77.8	71.7	50.9	21.1	47.6	54.3	62.6	56.8	70.2	75.9	74.5	68.9	71.7	53.7	71.4	72.1	60.1	28.4	78.8	20.6	44.6	70.9	53.0	35.9	69.0	53.2	98.9	58.9
gen_f1	74.2	58.0	68.5	59.6	63.9	57.2	80.0	83.4	79.8	75.5	82.5	84.9	52.2	76.2	77.4	73.7	52.3	77.1	80.1	68.9	79.1	69.3	66.7	75.8	76.7	55.2	83.3	78.9	74.2	54.8	58.1	63.1	70.9	65.9	78.1	80.9	80.1	75.6	80.3	68.2	78.0	76.8	71.9	36.8	84.1	60.5	53.7	79.0	62.5	56.0	75.1	70.8	99.4	70.5
gen_em	67.4	43.1	58.7	46.6	49.1	21.4	72.1	77.9	74.7	69.0	76.9	81.2	43.2	68.8	69.6	64.5	39.6	70.9	70.3	63.5	71.7	41.8	47.0	69.2	67.8	21.0	78.2	72.0	50.9	20.9	47.0	54.4	63.0	56.6	71.0	76.0	74.1	68.9	71.7	54.1	71.7	71.4	59.9	27.6	78.8	21.6	44.3	70.7	53.6	36.3	68.9	53.5	99.1	58.9
loc_f1	10.9	4.3	7.3	4.8	4.6	17.0	10.9	6.7	11.9	5.1	12.3	11.9	5.4	12.0	8.5	5.2	4.3	7.0	11.1	4.5	9.1	13.8	4.9	7.6	9.6	17.0	11.8	11.4	6.0	23.9	5.1	2.6	5.4	5.7	11.2	12.8	8.4	12.9	10.9	7.7	9.0	9.1	6.3	2.5	10.3	19.2	6.7	9.2	4.5	4.2	10.9	7.3	18.4	8.9
loc_em	7.0	0.9	3.9	1.6	2.0	0.9	6.6	4.7	7.7	3.1	8.3	7.5	2.2	6.7	5.3	3.0	1.4	4.9	6.6	2.6	5.5	1.9	1.4	4.3	5.5	0.1	7.6	7.3	1.5	0.7	1.5	2.0	2.8	2.7	7.3	8.6	5.1	7.8	7.6	4.0	5.8	5.7	3.4	0.7	6.9	0.5	2.8	5.6	2.6	1.1	6.1	1.5	13.2	4.1
port_f1	19.6	12.5	14.6	14.9	16.3	23.2	25.5	9.9	21.6	8.9	21.0	20.4	13.0	26.6	13.0	7.0	13.0	10.7	24.8	9.7	23.4	23.0	13.1	15.2	14.2	23.5	18.6	24.7	16.9	27.9	9.1	9.2	8.6	8.7	16.0	22.0	15.5	24.0	22.9	19.9	16.5	14.4	12.1	5.6	18.3	24.1	8.9	15.5	15.0	9.2	15.5	16.8	34.6	16.4
port_em	12.4	5.4	7.8	7.1	8.5	1.6	17.8	5.7	14.4	4.2	14.1	14.4	6.3	18.4	7.0	2.6	6.3	5.8	17.6	5.1	15.2	3.3	5.5	9.8	8.4	1.1	11.3	17.0	7.0	1.1	3.8	5.1	4.3	3.3	9.3	14.8	9.7	15.9	15.7	11.4	10.9	8.6	5.9	2.6	12.3	0.7	3.5	9.8	10.0	2.4	9.3	8.2	26.1	8.5

8a-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	77.6	59.8	73.7	64.3	66.0	63.1	82.0	86.4	79.1	75.2	84.5	84.7	52.8	78.6	78.2	72.5	52.8	78.1	80.5	72.2	77.7	67.5	67.9	76.1	78.1	29.0	82.5	80.0	74.4	57.6	65.2	62.6	68.3	53.4	82.6	80.6	79.1	78.2	82.0	67.8	78.3	78.9	72.3	43.4	80.8	61.2	57.6	80.2	68.6	55.8	75.4	64.0	99.4	70.9
rel_em	70.7	44.0	64.7	47.9	48.9	26.1	73.9	81.6	74.2	69.0	79.0	80.2	44.2	71.5	70.0	65.1	39.2	72.7	70.9	66.0	68.5	39.1	48.5	70.0	69.0	21.3	77.8	72.8	55.9	22.0	49.4	54.1	58.4	41.7	75.2	75.2	71.9	71.5	72.9	53.7	72.9	73.9	59.9	36.7	74.6	21.0	48.5	72.4	55.1	35.8	71.3	48.7	99.1	59.6
gen_f1	77.1	59.9	73.3	64.2	66.0	62.9	82.1	86.4	79.5	75.8	84.0	84.9	53.4	78.3	78.3	72.6	53.1	78.0	80.4	72.3	77.5	67.0	68.1	75.9	78.1	29.1	82.6	80.2	73.8	57.0	65.3	62.7	68.5	53.3	82.0	80.8	78.7	78.3	81.9	67.4	78.2	78.4	73.0	44.6	80.9	61.1	57.3	79.4	68.2	56.1	75.8	64.2	99.4	70.9
gen_em	70.3	44.0	64.3	48.5	48.6	25.7	73.9	81.6	73.9	69.3	78.3	80.4	44.4	71.2	70.4	65.6	38.9	72.4	70.8	66.0	68.1	38.6	48.5	69.3	68.7	21.1	77.6	73.1	55.1	21.6	49.8	54.3	59.0	41.1	74.9	75.7	71.5	71.5	73.0	53.4	72.5	73.5	60.5	37.1	74.7	21.0	48.5	71.8	54.7	35.9	71.3	48.8	99.1	59.5
loc_f1	13.6	7.3	12.3	6.6	8.5	20.3	14.8	12.8	14.0	10.8	14.6	14.3	8.5	15.3	11.6	11.3	5.2	10.8	15.7	8.7	13.5	16.2	9.5	11.9	13.6	20.3	15.3	15.5	10.4	22.5	7.2	9.3	8.4	6.7	13.8	15.8	12.9	14.8	15.2	10.1	12.1	11.4	11.2	6.0	15.5	21.5	8.9	14.0	8.4	7.2	12.7	7.7	20.6	12.2
loc_em	9.7	3.0	8.6	2.2	4.2	1.6	10.1	9.0	9.8	7.4	10.0	9.7	4.9	10.5	7.9	7.9	2.2	6.5	10.6	6.1	9.6	2.6	4.3	7.8	8.0	0.9	10.7	10.8	2.7	1.9	3.0	6.9	3.8	2.8	9.2	11.6	8.6	9.8	11.2	5.9	8.4	7.8	6.6	3.1	11.5	1.2	4.9	9.4	4.7	2.0	9.0	1.8	14.3	6.6
port_f1	27.8	12.9	22.2	18.8	22.7	27.9	33.1	18.2	31.7	13.8	30.8	30.5	14.8	35.4	22.1	12.6	17.1	24.7	35.0	10.7	30.3	22.3	18.5	23.2	20.7	25.9	29.4	32.0	19.6	31.3	12.5	16.5	16.6	12.5	22.4	33.1	24.6	32.8	32.2	26.4	23.7	23.3	15.6	9.3	27.7	28.4	14.4	25.5	23.1	13.1	24.3	20.8	50.8	23.0
port_em	19.7	6.7	14.8	10.1	13.1	3.2	25.2	13.1	22.9	8.1	23.0	21.5	8.2	27.3	14.7	6.9	8.6	17.4	25.8	5.9	21.4	5.1	8.1	15.9	12.8	3.0	22.4	23.6	7.9	2.8	4.7	10.9	10.5	6.4	15.9	24.9	17.8	24.6	23.6	17.9	16.3	16.2	9.3	5.0	19.9	1.4	6.9	17.6	15.2	3.8	18.2	11.5	41.2	13.8

Llama3.1-8B

0-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	76.0	51.6	70.7	50.6	57.3	41.7	68.6	86.1	77.2	70.3	81.0	75.6	47.4	73.6	71.9	65.3	39.8	79.0	76.4	67.8	65.9	62.1	63.9	74.3	72.8	43.7	86.4	74.2	50.0	47.1	50.5	65.7	71.0	68.4	77.8	79.7	74.1	77.8	77.0	62.0	74.5	70.4	71.9	57.5	81.7	53.3	47.3	66.1	56.7	43.4	69.6	43.0	96.9	65.5
rel_em	70.5	41.9	65.0	38.0	45.8	17.1	60.2	80.8	71.3	64.5	75.4	68.5	40.2	65.8	65.1	56.8	29.5	73.1	68.2	63.7	56.8	35.9	49.2	68.2	64.8	23.0	84.2	68.0	37.7	20.5	40.4	57.4	64.2	61.0	73.7	74.7	69.0	72.3	69.8	52.5	69.1	64.8	66.1	47.7	77.3	22.8	38.2	59.4	48.9	32.0	63.5	35.3	96.1	56.3
gen_f1	72.4	49.4	69.0	50.5	55.6	41.5	66.2	84.2	74.8	67.7	79.1	73.5	45.8	72.0	70.1	65.3	38.1	76.5	74.3	66.7	64.3	61.9	63.0	71.8	70.7	43.8	84.6	72.7	48.3	47.7	49.9	62.7	68.9	67.1	77.0	77.3	73.3	75.4	76.4	60.3	73.0	68.7	71.3	56.9	80.0	53.8	47.1	65.1	56.2	41.9	67.2	41.5	87.7	64.1
gen_em	66.9	39.6	63.4	38.2	43.9	16.4	56.9	78.1	68.4	61.1	73.5	66.1	38.9	63.9	63.3	56.4	28.1	70.5	66.9	63.3	55.6	35.2	47.8	65.4	62.5	23.3	81.7	66.5	35.3	20.2	39.8	54.4	62.2	59.5	72.5	71.7	68.3	69.3	69.1	50.7	67.4	64.0	65.3	46.8	75.7	22.9	38.2	58.4	47.8	30.9	60.7	33.2	85.3	54.7
loc_f1	9.6	4.7	10.2	4.9	6.8	17.8	11.3	7.2	10.3	6.7	13.0	12.3	5.9	11.8	6.4	6.3	4.9	8.5	11.5	4.3	9.7	14.3	7.2	10.0	10.3	19.1	12.3	11.8	7.3	25.4	7.5	6.1	6.5	6.5	10.3	12.9	10.4	12.9	11.1	7.4	9.6	8.9	5.5	7.0	12.7	20.0	8.4	11.7	7.2	5.9	10.5	6.5	19.7	9.8
loc_em	5.3	1.1	6.6	1.6	3.0	0.9	7.4	3.9	6.2	3.9	8.9	8.2	2.0	7.4	3.6	3.6	2.0	5.5	6.7	2.4	6.2	1.4	2.7	6.2	5.7	1.2	8.0	7.4	1.9	1.8	1.8	3.8	2.7	3.0	6.5	8.5	6.8	8.5	7.2	2.7	5.9	6.1	2.8	4.0	8.8	1.4	4.7	7.2	3.3	2.2	5.8	1.6	12.7	4.6
port_f1	10.2	6.7	7.4	8.4	9.4	21.3	9.6	8.2	11.5	6.1	10.4	12.0	5.8	12.2	5.3	5.0	7.2	8.8	12.3	5.2	9.7	20.1	10.0	9.8	9.2	18.8	8.8	13.7	9.2	25.4	6.7	6.5	5.2	6.0	7.7	13.9	9.5	13.4	9.2	13.2	8.5	7.5	6.7	8.1	10.9	23.0	5.5	9.3	10.7	5.9	11.1	6.9	25.8	10.1
port_em	5.1	3.1	4.0	4.2	5.1	2.6	4.9	4.2	7.1	2.8	5.7	6.5	1.1	6.6	1.9	2.0	2.8	4.7	6.6	2.6	4.7	4.3	4.5	5.4	4.9	1.6	5.3	8.6	3.6	0.7	2.7	2.2	2.3	2.6	4.3	8.6	5.4	8.3	4.7	7.4	4.7	3.8	3.5	4.3	5.7	0.8	1.9	5.3	6.1	1.6	7.3	3.0	18.2	4.3

1-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	81.3	66.8	42.1	66.5	72.7	63.3	80.4	86.3	85.7	79.4	83.2	86.3	59.5	82.4	81.2	77.1	60.4	81.8	84.1	76.9	76.6	74.7	76.4	79.7	81.1	59.0	87.0	81.6	81.2	58.7	72.8	64.1	71.1	73.7	80.7	84.9	80.6	82.7	83.8	74.8	81.7	81.6	77.4	67.4	84.8	60.5	65.3	81.5	73.0	62.7	78.8	77.2	98.3	75.3
rel_em	75.9	52.6	32.7	50.9	57.2	27.9	73.1	80.0	80.8	75.0	77.8	82.1	49.0	75.8	74.6	68.2	46.2	76.5	75.9	73.0	68.6	46.2	58.0	73.4	73.5	28.3	84.2	75.5	69.6	24.3	58.4	56.7	61.0	66.2	74.7	80.7	75.4	76.7	76.8	63.3	76.6	76.3	67.2	54.9	79.9	23.1	53.0	73.9	59.8	43.6	72.6	62.3	98.0	64.2
gen_f1	80.7	66.9	39.6	66.3	72.6	62.6	80.2	86.5	85.4	79.4	82.6	86.5	58.8	82.2	80.9	77.2	59.3	81.2	83.8	76.7	75.9	74.2	75.8	79.4	80.8	58.5	86.0	80.8	81.0	58.7	72.4	63.8	71.5	73.4	80.4	84.7	80.8	81.6	83.7	74.4	81.1	81.5	77.1	66.6	83.8	60.2	65.8	81.4	73.4	61.9	79.3	75.9	97.9	74.9
gen_em	75.0	52.4	29.9	51.3	56.7	27.5	73.0	80.2	80.1	74.7	76.7	82.5	48.9	75.6	74.6	68.0	45.2	75.8	75.6	72.5	67.7	46.2	57.6	73.1	73.2	27.9	82.8	74.7	68.8	23.9	57.8	56.3	61.8	65.5	74.4	80.2	75.4	75.2	76.9	62.7	76.0	76.3	66.9	53.9	78.4	22.2	53.6	73.6	60.6	42.5	73.1	61.1	97.4	63.8
loc_f1	16.1	6.8	9.5	8.4	9.9	21.0	15.4	12.8	15.2	11.3	17.1	18.3	8.4	16.2	12.6	10.3	6.9	13.4	17.7	8.3	14.7	15.6	8.6	14.7	14.8	21.8	15.8	15.8	11.8	26.4	9.4	6.3	10.7	10.6	15.6	19.7	13.9	17.0	16.5	10.9	12.8	13.3	11.9	9.9	16.5	20.8	9.7	12.4	9.8	7.3	12.7	11.7	25.1	13.4
loc_em	10.2	2.6	5.3	3.8	5.0	1.9	10.4	8.3	10.8	6.7	12.1	13.1	3.9	10.8	8.2	6.9	2.8	9.0	12.1	5.1	10.1	1.9	3.0	10.1	9.0	1.9	10.7	10.5	3.5	2.3	3.1	4.6	5.7	6.4	10.5	13.6	9.5	11.0	11.9	5.9	9.2	8.9	7.2	5.5	11.5	1.1	5.1	7.9	5.1	2.2	8.1	3.6	18.0	7.1
port_f1	25.1	14.6	7.3	17.9	20.5	25.3	25.4	17.6	24.9	16.0	27.6	28.4	15.5	25.2	17.5	10.3	16.8	21.3	26.2	15.3	22.3	28.2	19.9	23.3	19.2	27.2	21.9	29.1	21.6	29.7	14.9	10.2	16.2	13.2	20.5	29.2	24.3	28.4	24.5	22.1	19.6	19.3	18.6	17.1	26.2	26.0	11.4	22.3	21.8	14.5	19.1	21.9	38.9	20.8
port_em	17.0	7.4	3.4	9.3	10.6	3.8	18.2	12.5	17.4	10.9	20.1	20.2	6.7	18.2	11.6	5.3	8.9	14.8	18.4	10.9	15.8	7.0	8.8	16.7	11.6	3.1	15.2	20.9	12.3	1.6	7.2	5.7	9.3	7.0	14.2	20.6	18.5	21.4	17.7	13.3	13.8	12.5	12.7	10.8	18.4	1.5	5.4	15.7	13.5	5.3	13.8	11.6	29.7	12.1

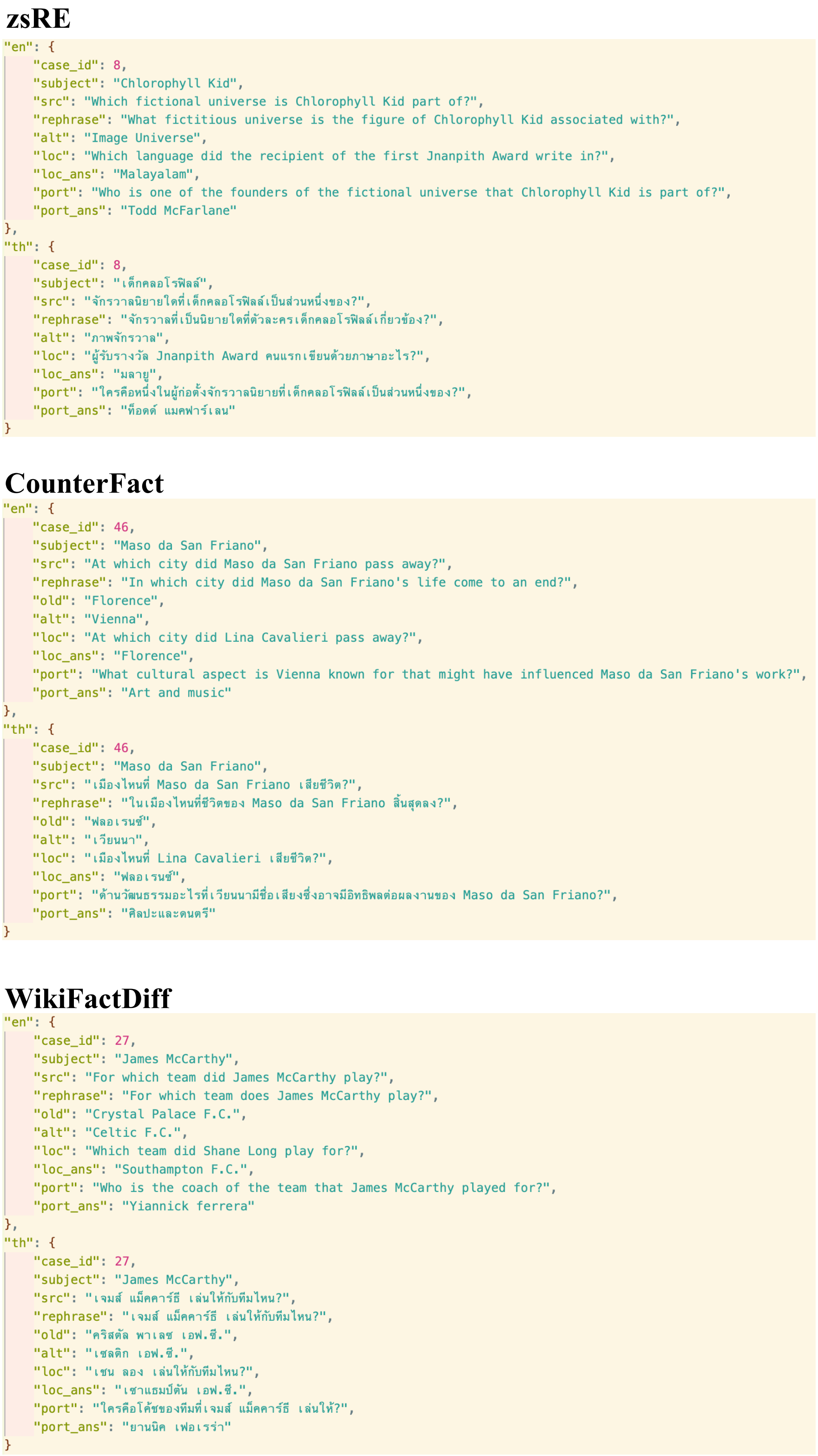
8-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	78.5	65.3	70.0	66.9	69.9	57.9	82.1	84.3	81.5	78.8	83.2	84.6	55.4	77.9	82.3	78.1	56.7	79.3	80.3	76.7	80.9	72.4	73.7	80.5	80.8	56.3	83.3	78.7	80.8	55.2	64.5	69.0	73.2	71.7	84.3	82.2	81.1	79.1	81.4	75.6	80.6	80.3	74.2	60.1	82.5	63.1	60.0	81.5	70.2	60.7	79.2	76.8	98.7	74.3
rel_em	72.3	51.4	59.0	52.4	56.1	20.3	73.5	77.9	76.9	73.6	77.1	80.9	47.9	70.0	76.6	66.9	43.3	73.8	70.9	72.4	73.5	45.7	56.0	74.6	72.2	25.6	78.0	70.8	65.8	23.9	54.5	60.9	64.0	64.1	77.9	76.2	75.8	72.0	73.3	63.7	74.1	75.5	62.5	48.8	76.6	24.7	52.4	73.7	60.2	40.8	72.0	62.4	98.3	63.2
gen_f1	79.0	64.3	69.7	66.0	69.5	57.5	81.8	84.7	81.5	78.7	82.4	84.4	55.5	76.9	81.2	78.2	57.2	79.1	78.9	76.1	80.2	72.4	73.6	79.8	80.5	56.5	82.6	78.7	80.6	54.7	63.5	68.8	72.0	72.1	83.4	82.0	80.6	78.5	81.4	75.1	80.5	80.4	74.8	58.9	82.8	63.2	59.1	81.9	69.9	61.5	79.3	76.3	98.5	74.0
gen_em	72.7	49.9	58.6	52.1	55.5	19.9	73.4	78.6	76.3	73.5	76.3	80.5	48.5	69.0	75.5	66.9	43.9	73.4	69.9	71.7	72.1	46.3	55.4	73.6	71.6	25.6	77.4	71.2	65.8	23.6	53.6	60.6	62.5	64.7	77.1	76.2	75.3	71.6	73.1	62.9	74.0	75.6	62.9	48.0	76.9	24.9	51.8	74.1	60.0	41.6	72.0	62.4	98.0	62.9
loc_f1	17.3	7.8	14.0	9.9	10.0	21.9	18.0	15.5	15.9	14.2	19.2	20.3	10.3	18.4	14.6	15.3	8.3	15.2	19.2	10.2	18.6	17.5	10.8	16.8	17.7	22.3	19.1	18.4	15.3	26.7	10.7	10.2	12.7	12.0	18.3	20.7	15.1	19.3	19.0	13.0	17.1	14.9	14.6	9.1	18.3	21.1	12.0	16.9	10.7	7.9	17.4	14.5	25.7	15.5
loc_em	11.3	2.4	8.8	4.2	4.9	2.0	11.8	10.8	10.1	9.6	13.6	14.3	5.3	11.8	9.0	10.2	4.0	10.6	12.7	6.9	12.3	2.7	4.7	11.0	11.1	2.3	12.8	12.5	6.3	2.3	4.5	7.2	6.8	6.5	12.5	14.2	10.1	12.1	13.1	7.3	12.0	10.0	9.5	5.0	11.9	0.8	6.6	11.3	5.1	2.4	11.2	5.4	18.4	8.4
port_f1	28.2	18.6	19.8	22.6	23.5	29.6	35.3	18.9	29.9	20.3	30.1	27.6	16.9	36.5	22.1	15.5	20.6	23.5	38.2	16.4	33.8	33.2	24.1	23.1	27.4	27.5	30.6	34.7	28.6	30.8	15.9	12.4	21.3	17.3	26.9	31.3	23.9	35.1	31.0	28.9	22.6	22.5	21.8	18.2	27.8	27.3	16.7	26.6	23.7	16.1	26.4	27.5	44.9	25.2
port_em	19.5	9.3	13.2	12.3	13.3	4.9	25.8	13.6	22.3	14.0	23.0	19.1	8.9	26.5	14.3	8.6	11.0	15.5	28.8	11.4	24.8	9.1	11.2	16.3	18.5	4.0	22.9	25.4	15.1	3.1	8.6	6.8	13.5	9.3	19.3	22.8	17.0	26.0	22.5	19.1	15.9	15.6	14.0	10.9	20.0	1.8	8.6	19.1	14.5	5.3	18.2	16.3	36.7	15.2

8a-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	79.0	66.3	76.0	67.2	70.1	63.7	83.6	85.6	82.8	80.4	83.7	84.2	60.6	80.1	83.2	76.0	58.3	80.1	82.1	79.0	81.7	74.1	75.8	80.5	79.4	42.0	84.0	80.8	81.2	59.7	70.5	65.4	71.8	60.9	85.5	83.4	79.7	79.3	82.3	72.7	81.8	81.8	74.7	65.7	82.1	64.7	60.7	82.6	72.3	61.6	78.5	74.2	99.2	74.9
rel_em	71.9	52.5	65.7	52.4	54.1	25.4	75.5	78.9	78.2	75.2	77.5	80.4	50.5	72.4	77.5	65.6	44.0	75.0	73.1	74.2	73.6	46.3	57.4	74.4	70.4	25.6	78.7	73.0	69.9	24.7	55.7	57.9	61.3	50.1	79.3	77.6	74.1	71.9	74.6	59.1	77.4	77.0	62.5	55.5	75.4	26.7	52.9	75.2	59.5	41.2	72.2	60.0	98.8	63.7
gen_f1	78.8	66.9	76.2	67.2	70.0	63.6	83.4	85.4	82.3	80.5	83.9	84.8	60.0	80.3	83.0	76.7	57.7	79.5	82.2	78.3	81.2	73.6	76.4	80.1	79.2	42.2	83.6	80.6	81.3	59.1	70.2	65.3	73.0	61.1	85.7	83.1	79.3	79.2	82.4	72.4	82.2	81.4	75.0	64.7	81.9	64.8	60.2	82.6	72.5	61.6	78.8	74.1	99.2	74.8
gen_em	72.1	53.3	65.8	52.5	54.1	25.2	75.1	78.9	77.0	75.2	77.8	80.8	50.7	72.8	77.3	66.5	43.9	74.2	73.0	73.5	72.8	45.5	58.1	73.9	70.2	25.4	78.0	73.1	70.1	24.0	56.0	57.6	62.6	50.1	79.3	77.0	73.3	72.1	74.4	58.8	77.5	76.4	62.9	54.2	75.7	27.2	52.5	75.3	60.0	40.9	72.6	60.2	98.8	63.6
loc_f1	19.3	8.7	13.3	8.3	11.1	20.7	19.2	17.7	17.5	15.7	20.3	21.3	10.8	20.0	15.8	16.6	7.0	16.6	20.3	12.6	18.9	18.4	13.0	16.7	18.4	23.1	21.0	19.7	14.2	24.9	7.3	11.6	14.3	11.6	19.3	22.5	16.3	20.0	18.1	14.0	16.6	17.6	16.3	11.6	19.6	22.4	12.3	17.8	12.2	9.0	18.8	10.1	27.0	16.2
loc_em	13.6	3.8	9.7	2.4	5.4	2.4	13.5	12.4	11.4	10.8	14.7	15.5	5.8	13.9	10.9	11.6	3.8	11.3	14.1	8.9	12.9	4.4	6.8	11.3	12.3	3.1	15.4	13.3	5.4	3.1	4.3	8.2	8.4	5.8	13.6	16.3	10.9	13.2	12.6	8.1	11.5	12.3	10.4	7.0	13.4	1.4	7.7	12.1	6.9	3.4	12.8	3.6	19.7	9.4
port_f1	38.7	21.4	27.5	28.0	29.3	33.6	40.2	30.2	38.8	26.7	41.7	44.2	22.3	41.2	33.9	23.4	23.7	37.0	42.6	24.8	39.4	33.9	28.8	36.1	36.0	30.9	41.8	41.6	34.0	33.8	25.5	21.4	26.7	24.3	37.4	43.4	37.9	42.3	39.7	32.7	36.2	34.5	27.7	24.6	39.2	31.0	22.2	37.6	33.1	20.4	32.0	33.5	55.1	32.9
port_em	28.4	12.8	18.6	16.0	18.3	5.7	30.2	23.4	28.8	20.2	31.2	33.8	13.7	30.2	25.2	15.5	12.8	28.5	31.8	17.6	29.6	11.2	13.9	26.4	26.2	5.7	32.8	31.6	19.0	3.5	13.8	15.3	17.5	15.2	29.1	33.3	28.7	32.6	29.3	23.7	27.4	25.7	18.4	16.0	29.5	2.3	12.5	28.5	22.9	8.4	23.7	21.0	43.6	21.5

Table 9: Full Results on zsRE.



zsRE

```
"en": {
    "case_id": 8,
    "subject": "Chlorophyll Kid",
    "src": "Which fictional universe is Chlorophyll Kid part of?",
    "rephrase": "What fictitious universe is the figure of Chlorophyll Kid associated with?",
    "alt": "Image Universe",
    "loc": "Which language did the recipient of the first Jnanpith Award write in?",
    "loc_ans": "Malayalam",
    "port": "Who is one of the founders of the fictional universe that Chlorophyll Kid is part of?",
    "port_ans": "Todd McFarlane"
},
"th": {
    "case_id": 8,
    "subject": "เด็กคลอโรฟิลล์",
    "src": "จักรวาลนิยายใดที่เด็กคลอโรฟิลล์เป็นส่วนหนึ่งของ?",
    "rephrase": "จักรวาลที่เป็นนิยายใดที่ตัวละครเด็กคลอโรฟิลล์เกี่ยวข้อง?",
    "alt": "ภาพจักรวาล",
    "loc": "ผู้รับรางวัล Jnanpith Award คนแรกเขียนด้วยภาษาอะไร?",
    "loc_ans": "มลายู",
    "port": "ใครคือหนึ่งในผู้ก่อตั้งจักรวาลนิยายที่เด็กคลอโรฟิลล์เป็นส่วนหนึ่งของ?",
    "port_ans": "ท็อดด์ แมคฟาร์เลน"
}
```

CounterFact

```
"en": {
    "case_id": 46,
    "subject": "Maso da San Friano",
    "src": "At which city did Maso da San Friano pass away?",
    "rephrase": "In which city did Maso da San Friano's life come to an end?",
    "old": "Florence",
    "alt": "Vienna",
    "loc": "At which city did Lina Cavalieri pass away?",
    "loc_ans": "Florence",
    "port": "What cultural aspect is Vienna known for that might have influenced Maso da San Friano's work?",
    "port_ans": "Art and music"
},
"th": {
    "case_id": 46,
    "subject": "Maso da San Friano",
    "src": "เมืองไหนที่ Maso da San Friano เสียชีวิต?",
    "rephrase": "ในเมืองไหนที่ชีวิตของ Maso da San Friano สิ้นสุดลง?",
    "old": "ฟลอเรนซ์",
    "alt": "เวียนนา",
    "loc": "เมืองไหนที่ Lina Cavalieri เสียชีวิต?",
    "loc_ans": "ฟลอเรนซ์",
    "port": "ด้านวัฒนธรรมอะไรที่เวียนนามีชื่อเสียงซึ่งอาจมีอิทธิพลต่อผลงานของ Maso da San Friano?",
    "port_ans": "ศิลปะและดนตรี"
}
```

WikiFactDiff

```
"en": {
    "case_id": 27,
    "subject": "James McCarthy",
    "src": "For which team did James McCarthy play?",
    "rephrase": "For which team does James McCarthy play?",
    "old": "Crystal Palace F.C.",
    "alt": "Celtic F.C.",
    "loc": "Which team did Shane Long play for?",
    "loc_ans": "Southampton F.C.",
    "port": "Who is the coach of the team that James McCarthy played for?",
    "port_ans": "Yiannick ferrera"
},
"th": {
    "case_id": 27,
    "subject": "James McCarthy",
    "src": "เจมส์ แม็คคาร์ธี เล่นให้กับทีมไหน?",
    "rephrase": "เจมส์ แม็คคาร์ธี เล่นให้กับทีมไหน?",
    "old": "คริสตัล พาเลซ เอฟ.ซี.",
    "alt": "เซลติก เอฟ.ซี.",
    "loc": "เชน ลอง เล่นให้กับทีมไหน?",
    "loc_ans": "เซาแธมป์ตัน เอฟ.ซี.",
    "port": "ใครคือโค้ชของทีมที่เจมส์ แม็คคาร์ธี เล่นให้?",
    "port_ans": "ยานนิค เฟอเรร่า"
}
```

Figure 3: Data Item Examples of BMIKE-53.

CounterFact

Llama3.2-3B

0-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	27.4	30.5	56.1	45.5	42.9	66.8	26.9	57.7	45.1	33.0	47.4	28.0	25.9	28.4	44.5	45.2	39.6	53.6	33.4	25.5	24.2	62.0	44.6	49.7	58.9	54.9	58.9	30.8	44.9	67.2	37.0	42.1	44.0	43.7	52.8	32.4	42.4	30.4	52.3	54.7	40.2	37.5	40.8	45.2	58.4	59.6	38.0	48.2	54.9	31.7	44.5	32.8	93.7	43.5
rel_em	25.0	21.6	52.5	34.3	32.7	40.8	22.9	55.5	41.7	30.8	45.2	26.2	17.5	25.6	39.8	38.3	35.5	45.6	30.5	23.1	21.2	31.8	37.3	46.2	51.4	26.8	56.0	27.5	39.3	33.5	32.3	35.9	37.4	34.4	49.9	30.6	38.7	27.1	48.9	50.1	36.3	34.0	36.6	40.6	56.2	26.9	29.1	43.5	48.6	23.5	43.5	28.8	92.9	36.3
gen_f1	26.5	26.5	52.5	44.7	42.2	66.1	25.3	55.3	36.9	34.1	45.5	24.7	25.0	17.9	39.4	45.4	39.4	48.2	34.9	25.6	20.9	58.9	44.4	44.8	51.8	54.2	57.5	33.2	43.6	66.1	34.3	43.0	38.4	38.6	48.6	28.3	35.8	24.7	46.4	53.1	37.7	34.5	40.0	39.8	58.4	58.6	34.2	43.4	49.3	30.3	43.3	31.9	79.9	40.8
gen_em	24.8	18.6	49.1	34.6	32.3	38.8	22.0	53.2	33.8	31.7	43.7	23.4	16.3	15.5	35.1	38.6	36.1	41.2	32.4	23.3	18.4	30.2	36.8	41.3	45.3	25.8	54.6	30.6	38.8	29.6	30.2	37.9	31.2	30.2	45.4	26.6	32.4	22.2	43.7	48.7	34.5	31.7	36.1	35.5	56.6	27.1	25.2	38.5	43.1	22.4	42.5	27.2	78.2	33.9
loc_f1	5.6	3.9	7.8	6.1	6.0	21.8	4.4	6.0	6.0	3.8	7.8	5.8	4.2	3.0	4.5	5.5	5.3	6.5	5.8	4.2	6.5	16.4	6.3	4.5	8.4	24.5	8.5	4.9	6.9	37.0	4.8	2.5	4.3	4.3	6.6	7.1	6.7	4.3	4.5	7.8	5.0	3.0	3.8	3.0	10.0	26.3	6.2	6.4	7.8	4.8	8.5	6.3	32.9	7.5
loc_em	4.7	1.5	6.5	3.3	3.1	6.1	2.9	5.3	4.3	3.0	6.6	4.5	1.2	2.0	2.9	3.6	4.2	4.9	4.3	3.7	4.7	3.0	3.5	3.2	6.4	1.4	6.8	3.2	3.3	3.1	2.6	1.7	2.6	2.2	5.0	6.2	5.1	3.0	3.1	6.0	3.1	2.0	2.5	1.7	8.2	2.6	2.9	5.0	5.2	2.4	7.2	2.9	31.8	3.8
port_f1	2.3	4.0	6.0	4.7	4.2	20.9	1.9	3.9	4.1	1.6	3.3	1.4	3.6	2.7	3.3	4.1	4.4	3.5	3.1	4.9	3.3	15.6	7.2	3.4	5.6	21.6	6.1	2.4	5.9	28.1	3.2	2.0	3.3	3.4	4.7	2.7	4.1	2.2	3.0	7.7	2.6	2.3	2.6	2.2	5.1	24.7	4.2	4.3	5.5	4.9	4.8	5.2	24.8	5.6
port_em	0.7	0.5	2.4	1.4	1.2	2.0	0.5	1.8	1.9	0.5	1.4	0.5	0.3	1.1	0.9	0.5	1.1	1.2	1.5	2.0	1.1	0.7	2.3	1.5	1.8	0.8	3.0	1.3	1.8	0.9	1.7	0.2	0.5	0.3	2.1	1.5	1.5	0.9	1.6	4.2	1.0	0.6	0.5	0.3	2.4	1.5	1.7	1.7	2.5	1.1	2.5	1.8	17.7	1.3

1-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	71.5	72.5	62.6	66.2	64.2	82.0	62.7	93.2	59.2	76.6	84.1	71.2	66.9	72.7	55.7	51.8	69.3	56.9	77.1	58.4	66.6	89.0	77.5	62.1	60.4	76.8	83.3	78.0	87.7	77.6	25.3	37.2	55.8	57.5	79.0	66.2	74.1	74.7	80.7	87.0	46.6	71.7	70.3	41.2	72.7	73.2	74.1	63.4	68.1	56.9	58.8	85.6	84.6	68.3
rel_em	68.4	63.3	54.2	54.8	52.7	56.5	59.0	92.9	54.7	74.8	82.1	69.1	56.5	69.2	39.7	31.5	64.7	40.3	74.6	55.1	61.0	79.4	69.2	58.1	44.8	43.6	80.2	75.3	85.2	31.4	24.4	30.0	46.0	49.7	76.1	64.1	71.0	71.1	77.8	84.4	41.4	68.4	64.9	35.3	70.5	37.5	68.4	49.5	61.3	43.3	58.3	82.0	84.5	59.9
gen_f1	70.4	71.8	61.1	65.9	67.5	81.3	60.8	93.3	59.8	76.6	82.1	71.9	66.2	71.3	55.6	49.0	69.0	57.7	76.2	57.1	64.8	88.2	76.5	61.6	61.2	76.6	81.8	77.0	87.6	77.6	24.3	39.3	55.7	57.3	78.3	64.6	72.4	74.2	80.1	86.5	48.1	71.4	69.5	44.8	71.9	71.9	73.6	64.4	71.3	56.7	58.4	84.9	89.2	68.0
gen_em	67.4	63.2	52.6	54.0	57.3	56.5	57.0	93.0	55.1	74.9	79.8	69.9	55.6	67.7	39.8	26.7	64.3	43.7	73.5	53.8	59.8	79.1	68.1	57.6	46.9	43.8	78.8	74.4	85.3	33.5	23.8	32.6	45.2	49.7	75.4	62.7	69.1	70.5	77.4	83.8	43.8	68.1	64.6	38.3	69.5	35.3	68.0	52.2	63.5	43.5	58.0	81.1	88.8	59.8
loc_f1	7.9	2.5	2.2	4.1	3.6	16.2	6.5	6.2	5.5	2.5	11.4	10.2	3.1	9.0	2.0	1.0	2.6	3.1	8.2	2.4	7.9	13.8	5.1	5.0	4.2	21.9	4.2	11.8	3.0	32.8	0.9	0.7	3.1	4.8	2.3	13.4	6.4	7.0	6.1	5.7	3.8	7.3	5.8	1.5	10.5	24.1	2.6	2.1	4.2	2.2	3.1	5.3	24.5	6.6
loc_em	7.3	1.2	1.8	1.9	1.4	1.8	6.2	5.8	4.6	2.3	10.9	9.4	1.7	8.5	1.1	0.6	1.8	2.7	7.7	1.9	7.2	2.9	3.0	4.0	3.5	0.5	3.6	11.2	1.0	0.3	0.4	0.3	1.1	2.7	1.9	12.9	5.6	6.1	5.5	3.8	2.9	6.4	5.1	0.5	9.8	0.5	0.8	1.6	2.5	0.2	2.7	1.8	24.0	3.7
port_f1	17.5	14.9	14.9	17.1	17.9	26.5	13.6	13.3	18.5	9.8	17.8	21.0	14.2	21.1	9.2	5.1	12.8	12.7	20.9	9.5	16.6	27.4	17.2	15.3	12.7	26.6	14.6	20.3	20.8	32.4	2.7	3.9	8.2	10.8	11.8	23.6	16.7	19.2	18.2	27.1	13.5	12.4	9.2	5.0	24.6	29.5	12.9	18.5	22.3	11.1	12.2	22.3	39.1	16.3
port_em	12.6	9.2	9.0	9.6	8.9	6.1	8.2	8.9	12.6	6.2	12.7	15.9	7.9	15.0	4.2	0.8	7.3	7.0	15.0	4.9	11.9	8.8	8.3	9.2	5.7	2.9	9.6	14.7	10.3	1.8	1.9	0.8	2.3	5.2	8.2	18.5	12.6	14.3	13.8	19.6	8.0	7.3	5.2	1.7	19.1	4.7	7.3	10.8	15.3	4.2	9.0	14.2	30.6	9.0

8-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	73.2	73.4	58.6	73.7	80.5	82.3	83.5	83.1	75.7	72.2	81.9	90.3	71.2	77.0	56.9	60.8	68.8	52.1	79.4	56.3	72.1	89.4	79.3	58.4	54.7	76.1	75.6	80.0	87.5	75.7	79.9	56.2	58.2	53.9	72.7	82.6	77.4	78.9	77.1	59.4	59.1	58.2	66.9	47.0	81.3	70.9	73.4	69.2	57.0	56.7	59.7	86.5	99.1	70.8
rel_em	69.5	67.3	51.0	65.3	74.6	56.7	81.3	81.0	72.5	69.5	77.6	89.4	64.6	72.9	41.0	41.6	63.7	35.0	76.1	50.2	67.0	81.4	70.2	46.5	37.9	47.2	72.0	76.6	84.3	32.4	75.6	48.5	46.7	43.6	70.6	81.0	72.4	76.0	72.7	51.1	52.9	54.8	61.1	43.2	77.7	36.7	69.1	57.4	48.1	43.3	58.7	81.4	99.0	62.3
gen_f1	73.5	73.1	58.2	72.9	79.8	81.2	83.1	82.2	73.5	71.7	81.5	89.7	71.0	76.1	56.2	60.6	68.7	53.1	78.7	53.7	70.9	89.3	78.7	59.5	55.4	75.2	75.7	79.3	87.3	76.5	79.2	56.0	55.9	53.3	72.5	81.4	76.6	77.3	76.0	59.5	58.1	58.7	66.0	47.5	80.3	70.3	72.0	68.9	59.0	56.7	59.7	86.2	97.9	70.3
gen_em	69.9	67.0	51.1	64.7	73.6	55.1	81.0	80.0	70.2	69.1	77.3	88.8	64.9	71.9	39.5	41.9	64.0	38.5	75.6	48.0	65.8	80.9	69.6	49.0	39.1	46.5	72.2	75.9	84.2	34.9	74.9	48.5	43.7	42.6	70.3	79.7	71.7	74.2	71.6	51.3	52.5	55.0	60.1	43.4	76.8	35.4	67.6	57.1	49.5	43.1	58.7	81.2	97.9	61.9
loc_f1	2.3	2.1	0.3	2.7	6.3	15.5	6.0	2.7	4.1	1.1	10.7	9.5	4.9	6.9	1.9	1.0	2.3	2.7	2.4	0.5	5.8	11.6	3.4	2.7	1.5	21.2	2.6	6.0	4.9	32.0	2.1	0.8	3.8	1.9	3.7	5.2	2.0	4.5	5.5	2.5	2.1	2.7	2.2	1.3	8.4	23.5	2.5	1.3	2.5	2.2	1.7	6.4	5.6	5.1
loc_em	1.8	0.6	0.1	0.7	4.1	1.3	5.2	2.4	3.2	0.9	9.6	8.2	3.0	6.3	1.2	0.6	1.7	2.3	1.9	0.1	5.0	0.6	1.3	1.4	1.0	0.0	1.7	5.1	3.2	0.1	0.7	0.6	1.2	0.3	2.6	4.8	1.1	3.8	3.9	0.8	1.2	1.6	1.7	0.6	6.8	0.2	0.8	0.9	1.1	0.1	0.9	3.0	5.0	2.2
port_f1	13.4	13.8	9.3	14.7	19.6	25.8	16.0	9.9	18.7	6.7	16.6	18.1	17.4	12.9	10.3	5.8	11.1	8.8	11.7	6.6	13.3	23.0	13.5	11.3	10.0	26.8	9.3	13.2	20.0	31.3	12.6	3.6	7.8	8.4	6.9	14.5	9.4	15.3	16.0	16.4	14.2	9.0	6.9	8.1	17.6	29.0	11.9	13.8	16.3	10.8	7.3	20.6	25.1	13.7
port_em	9.1	7.9	4.2	8.6	12.7	5.5	11.7	5.1	12.6	3.3	11.2	12.9	10.7	8.6	4.6	0.9	6.3	4.3	8.3	2.2	8.2	6.7	5.9	5.7	3.0	4.2	4.9	10.4	9.9	2.1	6.2	1.2	1.8	2.5	3.9	11.1	6.0	11.2	11.6	9.8	7.7	4.5	2.2	4.0	12.9	4.3	6.1	8.2	10.2	4.4	5.0	12.2	19.9	6.9

8a-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	68.0	59.5	62.8	65.3	64.4	80.7	79.1	65.0	56.7	70.1	79.4	88.1	79.1	78.3	58.8	59.1	66.1	53.1	72.0	59.0	66.7	89.0	77.9	55.3	67.5	74.0	80.2	67.3	83.9	76.5	77.3	49.8	51.0	51.9	72.5	81.9	52.3	45.7	78.4	61.6	55.1	58.0	65.3	44.7	78.0	73.3	76.5	77.2	66.1	59.2	69.1	65.8	99.3	67.6
rel_em	63.3	50.9	54.9	53.4	53.5	51.3	76.0	63.8	51.9	66.5	75.4	85.0	73.7	74.4	41.1	42.5	62.0	36.0	66.5	54.1	60.6	72.8	68.1	48.2	62.7	44.0	76.0	62.0	82.0	39.2	74.5	44.3	35.0	39.0	69.9	78.1	45.9	39.0	73.5	52.9	49.9	53.0	58.8	39.1	74.6	37.4	71.8	71.5	56.7	45.1	65.2	60.3	99.2	58.6
gen_f1	66.2	57.5	62.2	66.9	65.8	79.5	79.1	62.8	57.0	69.6	79.4	86.5	78.7	77.9	59.4	58.3	65.0	52.9	71.8	57.7	64.6	88.7	77.6	55.1	69.6	73.5	78.7	67.2	83.6	76.7	76.4	49.1	48.2	51.3	72.4	79.6	52.1	46.1	77.3	63.4	54.9	57.9	64.8	47.1	77.5	73.1	75.9	76.8	66.2	59.8	68.5	63.6	95.9	67.1
gen_em	61.9	49.3	53.9	55.1	55.8	48.8	75.9	61.8	52.2	66.2	75.3	83.3	73.6	73.8	42.0	40.7	60.7	37.7	66.4	52.8	58.5	73.9	67.6	48.0	64.7	43.7	74.7	61.9	81.5	40.1	73.9	43.7	31.5	38.1	70.0	76.0	45.9	39.8	72.4	55.1	50.6	53.0	58.4	40.2	73.7	36.8	71.4	71.7	55.6	46.1	64.5	58.6	95.7	58.2
loc_f1	41.8	16.1	28.0	18.8	24.0	35.8	39.1	33.9	36.3	23.6	46.6	49.9	29.5	49.3	29.1	23.6	14.4	29.5	42.4	11.5	43.5	28.0	23.0	34.5	30.5	29.1	40.4	44.9	34.1	43.0	10.4	30.5	20.4	16.4	35.5	49.6	36.6	43.1	42.9	28.6	29.3	31.8	33.9	16.3	45.3	29.4	31.0	34.1	31.2	7.1	35.1	31.4	55.7	31.6
loc_em	40.0	14.0	25.5	15.4	20.2	21.3	37.5	32.7	33.9	22.6	44.9	48.6	26.7	48.1	22.5	20.3	13.3	22.5	40.6	9.2	40.5	15.7	20.4	30.7	24.1	6.3	37.0	43.3	31.9	9.0	8.6	28.0	13.3	9.8	33.5	48.7	32.9	40.1	39.6	26.8	24.9	28.1	30.5	14.4	41.7	4.5	28.0	33.0	27.9	4.7	33.4	27.3	54.6	26.9
port_f1	32.8	24.2	18.2	28.4	33.6	34.7	38.8	30.3	33.4	22.6	41.3	44.5	27.9	42.7	28.8	21.6	20.3	32.4	42.4	17.6	40.4	34.1	26.4	32.2	32.3	34.4	39.2	39.8	35.8	38.5	24.2	21.3	21.8	15.5	31.4	43.0	33.8	40.3	36.8	38.3	29.6	24.6	26.0	13.1	44.3	33.5	19.7	36.8	33.7	17.1	31.4	36.9	60.2	31.2
port_em	24.1	16.9	10.7	18.6	23.1	9.0	29.9	23.8	26.3	16.1	34.0	35.1	19.9	35.2	20.8	12.6	10.9	22.4	33.6	11.1	32.1	11.2	11.5	24.3	23.6	7.4	31.6	32.3	21.1	6.4	12.3	14.2	12.4	8.0	23.9	34.8	27.9	31.8	28.1	30.1	20.8	17.1	17.9	8.5	36.5	4.9	12.6	28.3	23.8	8.2	24.8	23.9	50.7	20.9

Llama3.1-8B

0-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	62.9	62.8	65.9	59.0	67.9	68.8	52.9	87.6	64.6	60.7	75.6	68.5	56.8	60.1	63.5	56.3	61.3	62.4	63.9	55.3	51.6	79.3	66.9	67.4	68.2	57.7	75.3	63.9	55.9	65.9	65.0	52.9	58.4	62.0	73.2	71.0	60.4	64.9	72.3	74.4	58.5	59.7	62.6	57.2	75.6	69.1	40.9	58.7	64.0	43.8	58.5	44.5	94.4	63.0
rel_em	60.8	57.4	61.8	53.3	65.1	49.4	48.4	86.7	61.3	58.0	73.1	66.4	53.1	56.1	58.1	49.9	58.2	52.9	59.8	51.4	47.7	67.2	62.6	64.6	61.3	37.7	72.5	60.8	41.1	35.5	61.6	45.2	50.4	56.1	70.0	69.5	56.4	61.5	69.7	72.9	55.2	55.3	58.3	51.3	73.7	40.1	36.7	53.8	62.1	36.4	57.7	42.7	94.0	57.1
gen_f1	58.9	58.3	63.0	55.6	65.2	68.5	52.5	85.1	61.8	57.3	72.7	65.3	53.7	57.1	60.4	54.7	57.9	59.5	62.1	52.2	50.3	78.5	64.1	63.0	65.4	60.8	72.0	63.5	50.5	68.2	61.4	53.4	57.0	59.9	69.5	67.9	56.2	62.6	68.0	71.3	56.5	56.7	61.1	54.6	72.9	68.8	37.1	56.7	61.4	41.7	56.2	41.5	78.9	60.6
gen_em	56.5	52.7	59.1	50.2	62.1	47.3	48.0	84.3	58.1	54.9	70.5	62.7	50.2	53.1	55.2	49.0	55.0	51.0	58.4	48.5	46.4	65.9	59.6	60.1	58.5	40.7	69.3	60.1	37.1	36.8	58.5	46.3	48.7	53.0	66.6	66.3	53.0	59.3	65.4	69.7	53.3	52.6	57.2	49.5	71.0	40.0	31.7	52.3	58.9	34.8	55.4	38.6	77.6	54.7
loc_f1	15.3	10.1	17.1	17.0	18.1	27.4	17.6	21.1	18.7	11.7	28.3	20.1	12.8	24.6	15.4	10.2	10.2	21.0	21.1	12.8	17.3	22.6	13.8	18.4	22.9	31.0	26.6	17.8	14.0	40.6	13.5	7.7	14.7	14.8	21.4	23.1	18.9	19.4	23.3	15.9	15.6	14.2	16.9	15.7	30.5	30.2	11.0	20.7	19.3	9.2	25.9	14.0	39.4	18.7
loc_em	13.4	6.3	14.1	14.4	14.7	11.8	15.1	20.1	17.6	10.6	26.7	18.6	7.6	21.9	12.6	8.4	8.9	17.7	18.1	11.4	14.7	7.7	11.2	16.3	21.6	9.1	24.0	15.8	10.2	6.9	11.5	6.3	11.5	11.2	19.3	21.5	16.4	17.0	21.3	13.9	13.8	12.4	15.0	13.1	28.3	7.4	7.6	19.2	17.1	6.8	24.4	11.0	38.7	14.5
port_f1	10.3	8.8	7.4	11.0	10.0	23.4	9.3	8.1	12.3	5.9	13.4	13.5	7.1	13.6	5.7	4.6	9.1	9.5	13.9	7.7	10.2	24.0	12.8	10.6	9.2	22.0	11.8	12.5	8.2	29.8	7.8	3.9	6.0	7.1	10.2	16.0	10.6	11.4	11.3	16.2	8.8	6.7	6.5	8.6	15.7	25.6	7.3	9.1	12.9	5.5	10.9	6.5	25.6	11.2
port_em	5.7	4.0	4.0	6.9	6.2	4.3	4.1	4.0	7.2	2.6	8.6	7.8	2.9	7.4	2.0	0.9	4.7	5.0	7.9	3.5	4.5	4.5	6.9	5.9	3.9	2.3	7.1	7.0	1.9	1.9	4.2	1.2	1.8	2.8	6.1	10.0	6.2	5.7	7.7	11.1	4.8	2.3	2.9	4.6	11.1	1.7	4.8	4.6	9.0	1.5	6.2	3.7	18.5	5.0

1-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	75.3	77.8	59.3	70.3	67.7	85.2	69.6	93.6	50.9	77.5	87.2	77.4	85.0	81.9	57.7	43.4	80.7	52.5	81.7	62.0	67.3	91.9	84.9	52.3	54.9	73.9	83.6	83.4	95.6	79.6	40.0	33.5	72.7	66.9	81.4	77.6	74.1	76.7	81.0	91.6	40.4	74.5	69.1	50.3	86.7	78.0	80.2	68.8	77.6	65.6	56.7	92.3	94.8	71.9
rel_em	73.2	71.0	48.7	55.2	56.0	63.7	66.9	93.2	46.4	75.5	85.1	75.8	80.8	79.3	39.6	18.1	77.8	31.7	79.3	58.0	63.4	81.8	80.3	48.5	31.5	35.4	81.3	81.5	94.5	39.3	39.5	26.3	66.2	60.9	78.4	76.3	71.0	73.6	78.8	89.7	34.8	71.5	65.7	43.8	85.3	45.8	75.4	50.6	71.1	55.4	56.4	90.0	94.8	63.8
gen_f1	73.3	77.4	56.8	71.1	70.6	85.0	67.5	93.4	50.0	77.1	85.1	76.3	83.8	79.2	58.8	44.6	79.7	50.9	78.7	61.7	64.4	90.7	85.0	52.8	52.6	75.1	81.6	81.5	95.2	79.6	38.1	36.1	70.4	65.8	80.1	77.2	72.9	75.9	78.9	90.9	43.3	72.3	68.7	51.4	85.1	77.3	80.6	71.7	78.3	65.1	56.0	91.5	93.5	71.3
gen_em	71.2	71.0	46.3	55.5	59.9	63.4	64.8	93.0	45.4	74.9	82.9	74.7	78.9	76.3	40.6	20.7	76.5	31.3	76.2	58.1	60.0	80.1	80.7	49.5	30.5	38.8	79.4	79.5	94.1	40.7	37.7	29.7	62.6	59.1	77.1	76.0	70.1	72.9	76.5	88.8	38.5	69.0	65.3	45.4	83.5	45.3	75.3	58.0	70.9	53.6	55.7	88.9	93.3	63.4
loc_f1	17.8	7.6	7.2	8.6	11.8	21.7	17.2	10.4	12.6	7.3	16.5	14.6	11.4	12.2	10.4	5.4	10.8	9.4	12.1	11.7	9.8	25.9	10.3	8.1	9.9	26.2	12.7	17.9	11.2	36.9	2.0	5.3	9.6	6.2	10.0	16.8	16.4	13.1	10.7	10.6	6.6	14.0	10.9	8.5	13.7	29.5	16.2	9.8	12.2	6.2	10.7	13.9	29.8	12.7
loc_em	17.0	6.2	5.7	5.6	8.4	7.3	16.4	10.1	11.3	7.0	15.7	13.8	9.9	11.5	8.0	4.0	9.7	6.4	10.9	10.8	8.9	14.6	8.1	7.0	7.8	2.8	11.5	17.4	9.4	3.3	1.8	4.0	6.6	3.9	9.4	16.2	15.4	11.6	10.1	8.7	5.2	13.1	10.4	6.8	13.0	5.7	13.5	8.7	9.7	3.6	10.5	10.0	29.4	9.3
port_f1	24.6	22.1	17.2	21.1	28.6	34.0	20.1	16.0	24.7	12.0	28.5	28.0	22.1	25.4	15.9	8.7	21.7	17.4	20.0	13.8	20.5	34.8	26.5	21.0	15.8	30.5	23.4	26.1	28.8	36.3	3.1	8.7	17.4	14.5	18.2	29.3	21.3	23.6	22.2	31.1	17.4	18.5	15.6	13.1	28.7	32.6	21.3	20.6	30.5	18.3	19.0	30.1	44.3	21.9
port_em	17.4	14.9	9.3	10.8	16.8	9.7	13.8	11.1	19.4	7.8	22.0	21.3	14.7	18.4	8.7	2.6	13.7	8.7	14.3	8.6	13.5	11.5	15.5	14.9	6.7	3.4	17.3	19.6	17.2	3.5	1.8	4.1	8.3	7.9	12.8	23.1	17.1	17.4	16.2	23.1	12.2	13.3	10.6	7.5	22.5	5.5	13.7	12.8	21.6	7.8	14.1	20.5	34.6	13.1

8-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	75.9	79.2	59.7	78.0	82.9	85.0	84.3	83.1	76.2	71.3	79.4	88.5	78.7	78.8	60.2	70.8	79.4	0.0	85.2	61.6	76.7	92.7	83.9	69.4	67.3	72.7	78.1	80.9	95.5	79.4	88.2	60.4	68.2	61.8	76.6	88.5	78.5	78.5	74.0	76.0	65.7	70.4	65.6	68.9	83.8	79.0	84.2	73.9	80.5	68.4	69.8	92.0	96.4	75.1
rel_em	72.6	74.0	52.5	69.8	76.1	64.7	81.5	81.6	72.3	67.8	75.9	87.7	74.0	74.3	46.1	63.0	76.6	0.0	82.9	55.8	72.1	85.3	79.8	65.2	54.3	42.7	74.2	76.6	94.6	41.9	85.2	52.7	59.6	52.1	73.0	87.8	73.8	75.0	70.2	69.8	60.6	66.1	58.5	64.2	80.1	47.7	79.8	65.7	74.0	58.8	68.2	89.6	96.0	68.2
gen_f1	75.8	78.9	58.4	77.2	80.8	84.3	83.7	81.7	74.9	71.0	78.0	87.7	78.6	78.0	59.8	70.4	78.3	0.0	83.9	60.5	75.8	92.5	82.7	68.6	67.0	72.5	77.7	81.0	95.0	78.9	87.6	59.0	66.1	61.5	77.2	88.7	77.8	77.5	73.9	74.8	63.9	70.5	64.8	66.5	83.2	77.9	83.7	74.4	78.6	68.3	69.7	91.3	94.9	74.4
gen_em	72.6	73.2	50.9	68.6	73.4	63.8	81.0	80.1	70.5	67.1	74.4	86.8	74.0	73.4	45.2	63.1	75.8	0.0	81.5	54.6	70.6	85.2	78.5	64.3	54.4	42.5	73.9	76.6	94.0	41.0	84.7	51.6	57.3	51.8	73.8	88.1	73.0	74.1	70.0	68.3	58.8	65.9	58.0	61.8	79.6	47.3	79.3	67.3	71.8	58.1	68.2	89.1	94.2	67.5
loc_f1	9.3	5.6	6.5	8.6	13.8	20.5	7.7	4.5	11.4	5.8	23.0	12.3	5.4	7.9	13.9	8.6	8.8	0.0	8.1	3.8	15.9	14.9	14.0	15.8	14.2	33.6	5.3	9.2	15.6	35.6	5.5	2.0	14.3	7.1	2.1	13.2	10.9	7.8	15.6	6.1	12.9	14.4	6.8	20.3	9.1	30.1	7.3	12.0	6.3	7.9	9.0	12.3	4.2	11.4
loc_em	8.3	3.7	4.7	6.2	10.5	5.8	6.4	4.0	10.1	5.3	21.1	11.0	4.0	6.8	10.9	6.6	7.8	0.0	7.2	2.6	14.5	3.9	11.6	13.6	12.3	10.6	3.8	8.2	14.2	2.3	3.9	1.7	9.9	4.2	1.2	12.7	9.1	6.0	13.5	3.6	11.0	12.6	5.5	18.1	7.1	5.9	5.2	11.1	4.2	5.7	7.2	8.9	3.4	7.8
port_f1	25.3	18.4	14.5	28.7	32.2	32.5	30.5	16.5	26.9	14.9	37.5	30.7	25.8	28.2	17.7	13.4	20.4	0.0	21.1	12.2	30.0	33.1	20.5	25.7	22.0	31.0	21.9	29.1	29.0	35.1	15.1	5.9	20.0	13.7	15.1	29.4	26.4	30.2	30.2	33.3	25.0	21.0	16.6	22.1	31.9	31.0	17.9	27.3	29.7	17.5	20.2	30.6	27.5	23.7
port_em	17.0	11.6	7.5	19.8	22.4	8.7	22.6	10.9	20.2	10.0	29.7	23.8	17.7	20.4	9.5	6.5	12.4	0.0	14.6	7.2	21.3	13.3	11.0	17.6	11.5	5.5	15.6	21.4	16.5	3.7	8.3	2.9	10.4	5.2	10.7	22.4	20.4	22.7	21.2	23.2	17.4	13.9	9.7	15.9	24.4	5.0	10.1	19.2	20.2	8.1	14.8	19.7	20.2	14.5

8a-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	70.8	71.3	61.0	75.9	74.2	83.5	78.6	82.8	63.2	69.3	82.6	90.8	85.5	80.6	55.6	61.3	77.3	56.4	77.9	55.5	80.8	93.9	86.1	56.0	80.3	77.2	78.6	73.7	92.7	79.1	82.2	56.2	53.4	48.9	81.8	84.8	66.5	74.0	82.8	73.2	61.8	63.0	67.8	69.3	85.7	79.2	84.3	82.7	63.5	69.7	74.1	84.7	99.6	73.9
rel_em	67.1	64.5	51.5	65.0	66.3	53.3	74.5	81.8	59.5	65.2	79.2	88.5	81.4	76.4	34.8	45.2	74.3	37.3	72.8	49.9	78.4	85.2	81.9	49.5	75.2	50.3	74.3	68.3	91.8	43.2	80.5	49.9	33.1	30.5	78.6	81.5	60.8	70.2	78.4	66.2	58.0	57.7	60.8	64.1	82.0	45.8	79.7	77.5	53.6	59.9	71.0	82.6	99.5	65.5
gen_f1	70.9	71.1	60.3	75.7	74.4	83.0	78.1	81.8	60.9	67.9	81.8	90.1	84.4	79.2	55.9	61.7	77.5	54.9	74.8	54.5	79.4	93.3	85.3	55.3	80.3	76.5	77.9	72.4	91.9	78.5	81.3	54.8	51.9	47.4	81.0	84.0	66.3	72.2	81.6	75.3	62.7	63.9	67.1	66.4	83.9	78.9	84.0	82.7	64.1	69.7	73.0	84.7	94.0	73.2
gen_em	67.3	65.1	50.9	64.9	66.7	52.4	74.2	80.8	56.9	63.4	78.3	87.7	79.9	75.1	35.8	46.9	74.6	37.8	69.7	48.7	76.6	84.1	81.1	49.2	75.3	50.0	73.8	67.0	91.1	42.6	79.6	49.1	30.8	29.4	77.8	80.7	60.5	67.8	77.3	68.5	58.5	58.4	60.3	61.2	80.1	46.1	79.4	77.8	53.2	59.8	69.6	82.1	93.7	64.9
loc_f1	54.1	33.4	45.0	39.5	40.7	46.1	47.6	43.8	45.3	47.1	57.4	60.6	46.9	59.3	50.1	45.4	37.7	37.3	56.6	34.7	49.6	44.3	44.4	52.4	46.5	44.1	51.2	58.4	49.4	52.5	47.8	41.5	37.8	35.3	53.2	61.3	52.2	53.5	56.6	49.5	43.9	48.9	46.4	39.4	59.2	42.1	45.3	54.7	47.6	34.0	48.2	53.3	62.1	47.6
loc_em	51.7	30.2	40.5	34.0	34.9	30.0	45.8	41.5	43.0	45.0	54.8	58.7	45.2	57.0	47.5	40.7	36.4	27.2	54.4	30.8	45.8	30.2	41.2	48.1	38.9	20.8	46.6	56.0	47.3	18.2	45.8	38.5	27.4	24.3	50.5	59.9	47.7	48.7	52.8	47.1	39.6	45.6	42.4	36.2	55.4	18.6	42.0	51.9	43.0	29.9	46.2	50.5	60.4	42.0
port_f1	45.0	36.1	26.1	38.7	41.6	41.1	48.2	43.7	45.2	35.2	49.8	54.6	37.3	53.0	41.7	33.4	31.9	44.1	52.0	30.1	52.7	47.9	37.7	40.7	42.0	38.4	44.8	52.8	47.0	41.4	32.4	26.0	34.4	29.9	40.8	52.2	47.4	50.7	49.8	48.9	40.1	36.8	35.9	26.1	51.7	37.9	27.6	44.5	42.2	28.9	40.5	42.5	63.3	41.2
port_em	31.4	25.5	15.3	26.4	28.6	11.3	37.8	37.6	37.3	26.1	39.3	43.2	27.1	42.4	31.7	22.5	18.8	31.8	40.8	21.1	43.1	20.6	20.2	32.1	29.7	8.4	37.2	41.9	28.2	6.2	20.3	18.5	23.4	19.9	30.7	40.7	39.9	39.6	37.7	37.6	31.0	28.0	27.4	18.4	40.4	6.9	18.5	33.5	30.5	14.8	31.6	29.0	53.9	28.5

Table 10: Full Results on CounterFact.

WikiFactDiff

Llama3.2-3B

0-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	34.5	27.0	83.6	38.7	43.5	45.5	65.8	84.7	79.2	62.8	49.4	54.3	35.5	51.9	80.0	78.4	26.3	83.3	49.0	71.8	41.9	41.4	47.5	77.7	86.3	37.6	69.2	77.5	25.6	33.0	33.2	70.7	81.3	73.7	79.7	68.0	77.8	56.6	81.9	52.6	76.0	61.1	66.5	68.9	75.7	41.1	53.6	82.0	44.2	30.8	74.7	33.6	100.0	59.0
rel_em	21.1	14.8	79.6	17.1	18.0	8.4	53.8	76.4	70.3	54.9	23.3	29.9	28.5	20.5	73.3	73.7	16.6	76.0	19.6	67.1	19.8	8.4	28.7	64.4	77.9	7.8	58.7	69.9	14.9	7.4	21.2	63.7	73.3	64.8	73.6	56.5	70.2	33.0	75.6	34.6	67.2	37.4	53.6	52.6	66.7	10.6	46.4	76.1	21.4	18.6	68.6	26.7	100.0	44.5
gen_f1	50.8	26.7	81.4	34.5	44.0	43.2	61.4	81.4	79.1	48.8	52.9	53.7	34.2	44.0	77.3	78.0	25.4	81.8	47.2	68.9	46.7	40.9	48.7	76.3	84.9	37.6	71.1	71.3	24.9	34.6	31.4	74.6	78.8	69.9	80.5	71.1	76.9	48.2	80.8	48.0	75.7	64.9	67.7	70.2	78.0	40.1	52.5	79.8	42.4	30.5	72.6	33.2	97.5	58.1
gen_em	32.4	14.8	77.0	13.7	20.9	7.4	51.0	72.2	70.4	40.3	33.6	35.3	27.9	22.8	70.7	72.8	15.9	74.4	29.0	63.9	31.4	8.5	31.3	64.8	77.1	7.7	62.0	63.0	14.2	7.1	21.0	67.7	70.7	60.8	75.0	62.4	69.1	35.5	75.3	29.1	67.9	46.0	57.4	58.0	70.4	10.1	45.8	73.7	20.8	18.8	68.6	26.3	97.1	45.1
loc_f1	3.9	4.5	6.7	5.0	3.8	19.5	6.4	7.1	6.8	4.6	3.9	6.4	3.5	5.5	5.1	5.4	3.3	5.4	4.7	4.7	5.0	16.7	6.9	4.9	6.4	22.7	6.9	6.0	5.0	23.2	4.2	4.6	7.4	6.3	7.3	4.2	6.5	4.8	6.6	4.9	5.5	4.9	4.5	4.6	5.6	21.9	4.5	6.3	5.4	4.9	6.7	4.3	16.3	6.8
loc_em	0.5	1.0	3.1	1.5	0.6	0.8	2.2	3.1	3.1	2.3	1.3	1.9	1.1	1.5	1.8	3.2	1.3	2.8	1.7	2.4	1.0	0.4	2.3	2.3	3.1	0.8	4.0	3.2	1.5	0.5	1.5	2.6	3.3	3.3	4.1	1.2	4.3	1.4	3.4	2.2	2.9	2.0	2.0	1.5	3.4	1.0	2.3	2.9	1.8	0.9	3.4	0.8	11.2	2.1
port_f1	1.0	0.9	2.2	2.1	0.9	17.3	0.7	1.2	1.4	0.8	0.7	1.0	2.1	1.5	1.4	1.5	1.6	1.9	1.2	1.0	0.9	15.3	2.4	1.3	1.3	19.8	1.7	1.4	1.0	25.8	1.0	1.3	1.6	1.2	2.5	1.3	1.4	0.9	2.0	1.4	1.1	1.0	1.2	1.1	1.1	22.5	0.8	2.0	2.3	1.7	3.0	2.4	5.1	3.2
port_em	0.5	0.0	1.0	0.6	0.0	0.1	0.1	0.3	0.6	0.3	0.1	0.1	0.8	0.1	0.3	0.4	0.1	0.5	0.3	0.1	0.1	0.0	0.3	0.4	0.1	0.3	0.6	0.8	0.0	0.0	0.0	0.1	0.3	0.1	0.5	0.5	0.5	0.3	0.3	0.3	0.3	0.1	0.4	0.1	0.1	0.0	0.0	0.5	0.9	0.3	1.8	0.1	2.9	0.3

1-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	79.6	10.4	86.9	44.1	61.4	53.7	81.9	91.6	81.7	82.2	85.9	89.1	23.5	85.9	86.8	80.7	12.2	83.2	88.2	81.9	83.1	33.8	35.0	81.9	70.1	10.3	87.4	84.2	6.6	40.0	35.7	81.7	87.1	79.9	84.2	89.5	83.0	85.9	85.5	60.2	81.0	79.3	82.0	67.4	90.0	23.6	55.2	85.8	56.7	38.8	81.7	35.2	99.9	66.8
rel_em	66.7	7.9	83.0	19.5	35.0	11.2	78.2	85.7	72.7	77.2	77.3	82.3	3.7	80.6	79.9	75.8	10.7	75.4	81.9	78.1	76.2	12.0	30.7	72.8	58.5	8.2	79.8	77.0	2.7	4.6	25.8	75.5	79.0	70.9	80.1	82.9	75.0	78.2	80.9	39.6	72.6	69.6	76.3	57.5	83.9	10.6	46.9	79.8	30.4	22.1	78.6	29.9	99.6	56.8
gen_f1	82.8	11.1	86.0	42.5	60.0	52.5	83.2	91.2	83.2	81.5	86.2	88.6	23.0	85.3	86.0	80.2	11.8	83.4	86.4	81.0	82.1	33.8	34.6	82.3	69.8	10.6	86.9	83.9	6.9	40.1	34.6	81.7	85.1	80.2	83.4	88.6	82.1	85.8	85.5	58.9	79.2	78.2	82.3	69.4	88.8	22.7	53.0	84.9	55.6	38.3	81.4	34.8	99.1	66.4
gen_em	74.0	8.3	82.1	18.8	34.4	11.5	79.7	85.3	75.0	76.0	78.1	81.6	2.6	80.4	78.8	75.4	10.5	76.0	79.6	77.3	75.3	12.8	30.7	73.9	59.4	8.3	79.2	76.3	3.1	4.5	25.3	75.5	77.1	71.2	79.8	82.3	74.2	78.6	80.9	38.3	71.6	69.0	77.0	59.6	82.7	10.6	45.4	78.9	32.1	21.1	78.4	29.2	98.9	56.7
loc_f1	5.6	1.0	6.6	5.3	6.1	22.6	6.7	7.4	5.9	5.6	6.0	7.8	2.5	6.3	5.6	4.5	0.8	5.0	7.0	5.6	6.4	12.0	3.1	6.0	4.7	2.9	6.7	6.3	1.2	25.5	3.9	5.0	7.0	5.6	5.6	7.6	5.6	7.9	6.9	6.3	5.1	5.5	5.3	5.0	6.2	9.6	4.5	6.3	6.1	7.6	5.2	4.4	11.6	6.4
loc_em	2.7	0.3	2.9	1.2	1.5	0.9	3.2	3.4	2.9	3.4	3.2	4.0	0.1	3.2	2.8	2.6	0.4	2.7	4.0	3.4	3.1	0.5	1.3	2.6	1.8	0.6	3.7	3.7	0.4	0.3	1.2	2.7	2.4	2.3	3.5	3.7	3.3	3.2	3.3	2.0	2.4	2.7	2.8	1.9	3.3	0.6	1.9	3.2	1.5	0.9	3.1	1.7	6.4	2.3
port_f1	2.0	0.1	2.3	2.4	2.2	18.1	2.4	1.9	1.7	1.5	1.8	3.3	2.3	2.8	1.9	2.0	0.0	1.8	4.9	1.4	1.7	10.5	0.7	2.4	1.4	2.0	2.3	2.9	0.0	26.8	1.0	1.3	2.3	1.6	1.8	3.9	2.2	2.5	5.0	2.7	1.7	2.0	1.4	3.8	2.0	6.3	1.3	2.2	3.6	2.3	1.9	1.5	8.5	3.1
port_em	0.4	0.0	1.5	0.1	0.6	0.3	0.3	0.5	0.8	0.4	0.8	1.4	0.4	0.6	0.9	1.2	0.0	0.5	1.3	0.3	0.5	0.3	0.4	0.9	0.3	0.0	1.5	1.3	0.0	0.4	0.1	0.1	0.9	0.5	0.6	1.5	0.9	0.6	0.9	1.2	0.5	0.9	0.6	1.0	0.5	0.0	0.1	0.9	1.4	0.0	0.9	0.3	4.5	0.6

8-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	90.1	17.3	86.5	38.4	32.0	57.3	81.8	93.7	81.2	81.3	85.2	82.7	40.5	79.4	85.6	83.8	11.5	80.8	82.0	77.8	77.5	26.8	49.4	82.7	89.0	17.5	84.4	78.5	14.5	39.9	16.1	80.3	85.3	81.6	85.2	86.4	80.3	77.2	81.9	49.4	82.8	80.6	78.1	42.7	85.6	34.1	47.9	86.6	38.9	37.3	79.5	32.3	99.9	65.0
rel_em	83.7	13.8	79.1	23.6	18.6	10.6	79.1	88.3	74.4	76.4	76.5	74.9	32.3	75.4	77.8	76.9	10.2	70.0	75.9	74.7	72.7	16.2	27.4	75.9	81.6	10.0	77.2	72.2	9.8	4.5	12.5	74.2	76.2	75.3	82.4	80.7	73.9	70.8	77.6	33.7	75.9	73.2	75.0	35.7	77.9	10.2	46.9	81.2	27.8	11.7	77.3	25.1	99.6	56.1
gen_f1	89.1	17.1	86.4	39.2	32.4	56.0	82.3	93.3	80.9	80.5	84.4	82.4	37.4	79.1	84.2	83.2	11.4	81.0	81.6	75.6	77.2	26.2	48.5	82.0	88.1	17.3	83.8	78.0	14.6	40.1	16.1	79.6	84.7	81.4	85.0	85.9	80.5	77.0	81.3	51.3	82.4	80.3	77.9	42.1	85.2	33.9	47.6	86.0	40.4	36.1	79.6	32.0	99.3	64.6
gen_em	82.0	13.8	79.9	23.6	19.4	10.6	79.3	87.9	74.1	75.6	75.5	74.6	26.0	75.1	76.2	75.9	10.1	70.0	75.5	72.7	72.5	15.7	27.2	75.1	80.8	10.1	76.7	71.7	9.6	3.8	12.5	73.5	75.8	75.0	82.3	80.2	73.9	70.7	76.8	34.6	75.5	72.8	74.6	35.0	77.6	11.0	46.7	80.6	28.3	11.5	77.3	25.1	99.1	55.6
loc_f1	8.2	1.9	6.0	5.3	3.9	22.4	6.4	7.3	6.0	5.9	5.7	6.4	2.6	5.3	6.5	5.3	0.8	5.3	6.0	4.9	5.0	7.7	7.3	5.3	6.6	6.2	6.0	5.8	2.4	24.9	1.5	4.8	6.9	6.0	5.7	6.9	5.4	5.0	5.9	4.6	5.6	5.3	4.8	4.3	5.5	17.9	2.4	5.6	5.5	8.1	5.1	2.8	10.8	6.2
loc_em	2.9	0.9	2.7	1.9	1.7	0.8	3.8	3.6	3.2	3.8	3.2	3.3	1.5	3.2	3.2	2.4	0.4	2.6	4.1	3.1	2.8	0.9	1.7	2.7	2.8	0.8	3.1	3.7	0.9	0.1	0.5	2.7	2.4	2.8	3.6	3.2	3.4	2.9	3.7	1.7	2.7	2.6	2.9	1.2	3.3	0.5	1.5	2.9	1.9	0.8	3.1	1.0	5.6	2.4
port_f1	4.6	0.4	2.8	2.3	1.8	19.7	4.4	1.8	4.1	1.8	2.4	4.7	1.6	5.7	2.3	2.4	0.2	5.0	5.8	1.2	2.5	6.7	2.9	4.5	2.4	6.8	2.7	6.6	0.2	27.0	0.3	1.3	2.0	1.1	2.5	6.0	3.5	6.5	4.5	3.8	2.6	2.9	1.6	1.5	3.9	13.0	0.5	2.9	3.7	2.9	2.7	0.9	10.5	4.0
port_em	1.7	0.1	1.7	0.4	0.6	0.5	2.6	0.5	1.7	0.8	1.2	2.7	0.5	2.7	0.9	1.7	0.1	2.0	2.8	0.3	1.2	0.7	0.8	2.3	0.6	0.0	1.3	3.4	0.0	0.4	0.1	0.3	0.9	0.1	1.3	3.1	1.2	2.7	0.6	1.3	1.4	1.7	0.8	0.4	2.7	0.0	0.0	1.3	1.3	0.0	1.5	0.1	6.1	1.1

8a-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	90.9	21.2	87.8	43.0	27.5	57.3	89.2	93.2	86.1	81.1	90.6	90.5	33.4	85.8	84.2	79.1	27.1	82.0	85.0	81.5	80.4	50.4	52.2	81.4	86.4	28.3	87.1	87.6	28.9	40.9	27.4	78.9	83.0	72.1	90.3	92.1	82.6	67.1	84.5	50.8	84.2	82.0	78.9	14.0	90.2	38.4	55.6	93.5	51.2	45.1	83.7	38.3	99.9	67.8
rel_em	84.4	15.2	82.7	19.5	18.5	9.2	85.0	87.2	77.2	74.5	83.0	83.4	11.9	80.1	75.8	73.6	21.7	72.8	76.0	77.8	74.7	20.4	38.7	70.7	79.7	9.8	78.8	79.1	18.2	2.3	20.3	73.3	71.7	62.0	86.2	84.7	76.0	43.1	77.8	33.3	76.8	73.3	66.5	9.6	83.3	5.4	52.9	90.0	30.2	18.0	80.4	30.9	99.6	56.3
gen_f1	90.0	20.8	86.6	43.1	28.1	56.1	88.4	92.9	85.8	81.0	89.7	89.3	32.7	85.2	83.4	78.1	26.2	81.5	85.4	81.1	80.1	49.4	51.4	81.5	85.5	28.2	86.3	86.2	27.9	41.3	26.3	78.4	83.4	71.9	89.7	91.5	82.1	70.8	84.4	51.6	83.6	81.8	79.3	16.4	90.1	38.8	55.4	93.1	52.4	44.4	83.3	37.3	99.3	67.5
gen_em	82.5	14.9	81.4	20.4	18.6	8.9	84.2	87.0	77.0	74.7	82.0	81.8	11.1	79.3	74.9	71.9	21.1	72.5	76.4	77.2	74.5	19.9	38.0	71.8	79.1	10.2	78.0	77.6	17.6	2.3	19.4	72.6	73.2	62.1	86.1	83.9	75.5	50.1	78.3	34.6	76.2	73.2	68.0	12.1	83.3	5.7	53.1	89.7	33.0	17.7	79.9	30.5	99.1	56.2
loc_f1	8.7	3.7	8.6	5.6	6.5	22.6	11.7	12.8	7.6	6.1	8.3	9.2	3.6	11.5	7.3	6.5	2.5	7.1	10.3	7.1	9.8	17.9	7.9	7.6	8.6	19.9	11.6	10.4	6.0	21.4	3.4	8.7	10.3	9.2	8.0	10.5	6.6	9.6	10.2	7.0	8.0	7.7	8.7	2.1	8.4	23.7	5.2	9.7	6.5	7.8	7.5	7.8	15.9	9.1
loc_em	5.0	2.4	5.7	2.4	2.9	1.0	7.3	8.2	4.3	3.8	5.4	5.6	1.9	7.3	4.2	4.0	1.7	3.7	5.7	5.0	4.9	0.9	1.2	4.3	4.2	0.8	6.9	7.9	2.4	0.3	1.8	6.0	4.6	4.4	6.0	6.3	4.5	5.0	5.7	2.6	5.2	4.9	5.6	0.4	4.6	0.4	2.7	6.0	2.3	1.0	5.0	2.7	10.5	4.0
port_f1	11.8	2.4	8.4	6.4	6.5	25.2	12.8	5.2	12.0	8.5	11.1	9.8	4.3	13.6	8.8	8.5	7.1	12.0	13.6	5.0	14.3	19.3	6.2	13.0	12.9	25.4	11.5	12.2	5.3	30.0	2.4	3.3	5.1	3.7	12.2	12.5	13.2	13.4	13.6	6.9	11.0	12.1	11.6	1.8	12.1	25.8	5.9	13.1	8.7	5.2	9.4	11.9	15.6	10.7
port_em	8.4	0.9	5.7	1.0	2.6	0.9	9.2	3.2	8.2	5.6	8.3	6.6	2.2	10.2	6.4	6.4	3.3	8.6	10.3	2.7	10.5	2.7	1.3	10.1	9.5	0.6	7.8	9.1	1.7	0.0	1.3	1.0	1.7	1.9	9.1	8.9	9.7	10.6	9.4	3.6	7.9	9.7	8.8	0.5	8.9	0.0	4.2	9.8	3.8	1.8	8.2	2.8	11.0	5.5

Llama3.1-8B

0-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	87.3	31.2	85.2	31.8	38.6	30.3	82.3	93.1	85.1	77.0	85.7	86.6	44.4	80.5	86.3	81.3	32.6	86.5	82.0	83.0	71.1	44.7	57.6	84.7	85.5	20.6	85.9	86.3	30.3	27.6	47.5	82.9	84.4	82.1	85.1	90.7	85.0	85.4	85.4	35.4	83.2	81.9	85.6	75.7	90.9	29.6	59.0	83.4	33.7	34.5	82.1	39.2	99.6	67.8
rel_em	79.0	20.7	82.1	21.7	23.9	9.2	78.6	87.5	77.9	70.8	76.0	79.0	35.6	70.4	80.4	76.2	24.0	80.5	71.4	79.3	53.4	18.8	42.2	77.6	79.1	10.1	78.4	78.3	15.4	10.6	33.4	77.7	78.4	74.1	81.4	85.2	79.2	78.6	81.5	24.4	75.8	72.8	80.1	67.7	85.6	13.8	53.7	78.7	24.7	23.9	80.1	32.7	99.5	59.1
gen_f1	85.7	30.1	84.0	29.9	34.7	29.8	81.2	92.2	84.0	71.1	84.3	85.6	42.8	77.3	84.9	81.0	31.0	85.3	78.4	80.6	74.8	45.0	55.6	83.7	83.4	22.0	84.2	83.4	26.3	28.2	44.1	82.7	85.0	81.8	82.9	88.9	83.1	85.8	83.9	33.5	83.9	80.5	84.4	75.6	88.7	30.5	57.4	80.8	31.5	32.9	79.5	35.3	93.7	66.4
gen_em	78.2	19.9	81.3	20.4	21.9	9.1	77.2	86.9	76.8	63.8	75.6	77.7	34.8	66.3	78.7	75.6	23.6	79.2	67.9	76.9	62.8	18.1	41.1	75.6	75.7	10.2	77.2	75.0	13.5	10.2	30.3	77.8	78.4	74.3	79.6	83.0	77.4	79.0	80.1	22.5	76.5	71.4	78.6	66.3	83.4	12.9	52.3	75.9	23.4	22.3	77.2	30.6	92.7	57.8
loc_f1	10.5	6.9	10.0	4.6	6.1	16.0	11.4	13.2	11.5	6.6	9.0	12.9	4.4	9.8	10.3	7.9	4.5	10.7	11.6	9.0	10.3	17.5	7.8	10.6	11.7	19.3	11.6	11.4	5.0	23.9	7.6	8.7	9.7	9.2	12.1	11.2	10.1	11.7	12.1	5.4	8.8	9.4	10.6	9.0	10.8	17.1	6.1	9.6	5.0	5.1	10.5	6.2	17.8	10.0
loc_em	5.4	3.2	5.4	1.8	2.6	1.3	5.7	8.4	7.5	4.6	4.9	8.0	1.4	5.4	6.1	4.7	2.7	7.1	7.3	6.0	5.0	0.8	2.8	6.5	7.4	2.2	7.0	7.3	2.2	1.7	3.5	5.2	5.0	5.4	7.3	7.8	6.3	6.4	7.8	2.6	4.9	5.6	5.0	5.1	7.0	0.9	2.4	4.7	2.4	1.7	6.8	2.3	12.5	4.8
port_f1	2.3	2.4	2.9	3.4	3.2	14.9	2.8	2.9	2.3	1.4	2.9	4.4	2.6	4.4	2.1	2.4	2.4	2.9	5.1	1.9	2.9	16.8	6.2	3.1	2.3	11.8	2.1	3.7	2.5	20.0	1.7	1.3	1.9	1.8	1.7	5.2	3.2	4.1	4.9	2.9	1.8	3.0	1.9	2.7	3.0	18.7	1.3	3.4	3.1	3.3	2.2	1.9	8.5	4.2
port_em	0.6	0.1	1.4	1.3	1.5	0.9	0.5	0.9	0.9	0.5	0.9	2.6	1.0	0.9	1.2	1.3	0.4	1.3	1.5	0.4	1.2	0.5	2.6	1.2	0.6	0.0	1.2	1.9	0.3	0.4	0.4	0.1	0.6	0.9	0.4	1.7	1.0	1.7	1.8	1.3	0.6	0.8	1.0	1.3	1.5	0.0	0.4	1.7	1.7	2.2	1.3	0.0	5.4	1.0

1-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	91.8	15.6	86.7	41.1	70.7	60.9	86.1	94.1	86.8	82.0	89.8	87.9	29.4	88.6	87.8	84.4	16.0	90.2	88.6	83.6	83.7	39.3	40.7	87.7	85.3	8.8	87.8	91.3	11.4	43.2	47.9	83.4	82.7	86.6	84.6	92.6	81.6	85.9	85.7	58.6	84.9	86.0	84.9	81.1	91.0	31.6	60.5	90.6	59.3	44.8	81.2	41.0	100.0	70.5
rel_em	85.6	12.6	83.6	25.5	46.9	14.8	82.7	89.0	79.3	77.2	82.8	80.0	4.6	83.6	82.4	79.3	14.5	84.3	82.7	80.4	76.7	19.9	34.7	80.1	79.6	8.2	81.1	84.6	7.4	8.3	35.2	77.8	71.5	80.0	82.1	87.1	75.4	78.2	81.5	41.9	77.3	78.3	79.5	70.9	85.1	13.4	55.0	85.8	34.4	23.2	78.6	33.9	100.0	61.1
gen_f1	90.2	15.3	86.5	40.8	69.3	59.1	85.6	93.0	88.4	82.2	88.2	87.1	28.8	88.0	87.7	83.8	15.2	89.5	86.8	82.5	82.8	38.5	40.3	86.3	83.8	9.1	87.7	88.2	12.0	43.6	47.0	82.9	82.8	84.8	83.3	91.3	82.2	85.8	84.0	57.5	85.2	85.4	84.9	81.1	90.4	31.2	59.7	89.7	58.6	43.5	80.3	39.9	99.5	69.8
gen_em	82.7	12.4	83.3	25.3	45.4	15.1	82.3	87.9	80.7	76.9	80.4	79.3	3.2	83.2	82.0	78.4	13.7	83.7	80.5	79.1	76.3	20.4	34.4	79.3	77.7	8.0	81.2	81.4	7.8	7.7	34.0	77.6	71.9	77.0	80.8	85.6	76.3	78.2	80.0	40.7	77.7	77.7	79.3	71.7	84.3	13.5	55.1	84.9	34.4	23.5	77.9	33.3	99.5	60.5
loc_f1	10.1	1.1	6.8	5.0	7.5	24.2	9.7	8.1	8.5	6.1	7.2	8.3	3.1	8.0	7.8	5.7	1.2	7.9	10.3	5.8	7.9	14.8	4.5	8.3	6.4	4.6	7.1	8.8	1.6	26.2	6.4	6.2	6.9	8.2	5.7	8.6	7.0	8.6	7.6	6.4	8.2	7.6	7.5	5.9	7.1	15.0	4.3	8.8	6.5	9.3	6.9	4.5	13.5	7.8
loc_em	5.1	0.4	3.1	1.7	2.6	0.9	5.9	4.0	4.9	3.8	3.8	4.7	0.1	4.5	4.0	2.9	0.8	4.5	6.0	3.6	4.1	1.3	1.7	4.7	3.5	1.0	4.1	4.7	0.8	0.3	2.2	3.6	2.6	3.5	3.8	4.9	4.5	4.5	4.7	2.0	4.3	3.6	3.8	2.7	4.5	1.2	2.2	4.9	2.1	2.3	4.3	1.5	8.0	3.2
port_f1	4.0	0.1	3.0	3.3	4.2	19.5	3.2	2.3	3.2	2.2	2.8	5.0	2.8	4.8	2.9	2.2	0.4	2.8	6.8	2.2	2.4	13.4	3.2	4.6	1.7	2.6	1.9	4.8	0.1	27.8	1.5	1.8	2.2	2.0	1.4	6.3	2.7	5.2	6.3	3.3	1.9	3.2	1.7	3.8	3.5	10.4	1.3	3.0	4.7	5.0	2.3	2.3	13.4	4.2
port_em	1.8	0.0	1.8	0.9	1.7	0.5	0.6	0.9	1.8	1.2	0.9	2.0	0.5	2.2	1.9	1.0	0.3	1.2	2.8	0.8	0.9	0.4	2.0	1.9	0.4	0.0	0.5	2.6	0.0	0.5	0.5	0.6	0.8	0.9	0.3	3.1	1.3	2.4	3.3	1.7	0.8	1.7	1.0	1.5	2.3	0.0	0.3	1.5	2.1	2.2	1.3	0.3	7.7	1.2

8-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	91.9	25.0	0.0	39.4	45.6	60.8	83.3	93.5	85.0	85.6	88.8	85.2	47.0	85.2	87.7	85.7	22.3	85.1	84.9	82.0	84.3	52.8	51.7	86.2	89.6	23.1	87.9	82.3	26.4	43.7	28.4	81.7	87.0	87.7	92.7	91.0	82.1	84.0	87.0	37.3	84.4	84.6	85.1	71.4	88.1	42.9	56.3	88.4	42.2	46.6	83.4	41.6	100.0	68.6
rel_em	84.4	19.9	0.0	23.5	29.2	14.2	80.4	88.1	78.4	79.6	81.1	77.6	37.8	81.5	80.6	76.9	20.0	68.9	78.6	78.7	79.6	24.1	29.5	78.7	82.9	11.1	81.1	76.0	18.1	4.7	21.9	75.6	78.3	81.4	90.9	85.2	75.4	77.3	81.8	26.4	77.8	77.0	79.0	61.5	81.1	10.3	53.3	83.4	27.8	15.9	80.2	32.4	100.0	58.4
gen_f1	91.5	24.0	0.0	39.2	45.0	58.7	83.4	92.8	85.5	85.2	86.9	84.9	45.8	84.2	87.1	85.5	22.0	85.4	84.3	81.2	83.5	53.9	50.7	85.5	89.3	23.8	87.3	81.6	25.7	43.0	27.8	81.3	86.4	87.6	90.8	90.5	81.6	82.2	85.1	36.8	83.9	84.0	84.5	65.6	87.1	42.2	55.0	88.1	42.3	45.6	82.9	40.7	98.9	67.9
gen_em	84.2	19.4	0.0	23.2	28.8	14.0	80.4	87.4	79.0	79.3	78.4	77.4	36.7	80.0	80.0	76.4	19.8	74.0	78.3	77.9	78.6	24.1	29.0	77.8	82.5	11.1	80.6	75.1	17.0	4.2	21.4	75.4	77.8	82.1	88.9	84.7	75.0	75.5	80.1	25.5	77.3	76.3	78.8	55.9	79.9	10.2	52.6	83.0	27.9	15.4	80.2	31.6	98.9	57.9
loc_f1	11.5	4.9	0.0	6.4	6.4	24.4	7.3	9.9	8.6	6.7	7.3	8.0	4.8	7.8	8.3	7.8	2.4	10.3	9.5	5.9	9.1	18.4	7.3	7.2	7.8	11.5	7.5	8.6	3.6	26.4	3.1	6.0	7.7	8.2	10.1	8.7	5.9	8.2	8.0	4.0	6.5	6.0	7.1	4.9	7.3	21.7	4.0	8.4	5.2	10.3	7.7	6.1	16.8	8.3
loc_em	5.6	2.3	0.0	2.0	2.9	1.2	4.7	5.4	5.4	4.5	4.6	4.5	2.8	5.2	4.6	4.0	1.8	5.9	6.6	3.7	5.6	1.2	2.4	3.7	4.3	0.9	4.5	5.5	1.2	0.6	0.9	3.8	2.9	3.5	6.3	4.7	4.0	5.0	4.7	2.2	3.2	2.7	3.8	1.4	4.7	0.5	2.3	5.1	1.8	1.7	5.0	2.6	12.0	3.5
port_f1	11.5	3.8	0.0	6.1	5.7	24.6	11.7	4.9	10.4	5.0	8.3	10.7	3.2	11.0	3.8	6.7	1.0	9.9	10.4	4.3	10.9	21.7	6.8	8.8	9.0	14.6	11.0	13.2	1.5	29.0	2.3	2.5	3.4	2.5	13.1	12.9	10.4	12.6	12.6	9.2	9.5	9.0	3.8	2.0	11.6	18.5	2.5	12.3	7.2	8.1	9.9	6.3	18.2	8.9
port_em	8.4	0.8	0.0	1.4	1.7	1.2	8.3	2.9	6.1	4.0	6.0	7.3	1.2	7.7	2.3	4.7	0.3	5.7	6.4	2.4	7.0	3.1	2.3	6.3	5.2	0.0	6.4	9.2	0.4	0.3	1.3	1.4	1.7	1.0	8.7	9.1	6.8	8.4	8.4	4.0	6.1	6.1	2.9	0.4	8.6	0.3	1.2	9.1	4.0	1.9	8.2	1.3	12.0	4.2

8a-shot

	af	ar	az	be	bg	bn	ca	ce	cs	cy	da	de	el	es	et	eu	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ka	ko	la	lt	lv	ms	nl	pl	pt	ro	ru	sk	sl	sq	sr	sv	ta	th	tr	uk	ur	vi	zh	en	Avg.
rel_f1	90.5	28.6	84.9	48.1	29.3	60.3	91.8	88.6	87.9	85.4	91.4	89.8	46.5	91.5	88.0	83.5	30.6	86.2	90.8	86.1	83.6	50.6	50.9	87.9	89.3	34.5	89.5	92.2	36.2	43.6	29.3	83.1	80.3	76.5	93.8	93.4	86.9	87.6	92.2	43.1	86.2	84.6	85.5	69.3	90.6	43.3	59.4	95.2	39.5	47.6	91.0	46.8	100.0	72.0
rel_em	83.2	21.8	76.4	26.5	20.5	13.5	85.1	80.2	79.1	76.7	84.7	81.4	36.4	85.3	80.5	78.4	24.2	76.0	82.7	81.5	76.8	20.2	31.1	78.7	81.0	10.1	82.0	85.0	23.3	4.2	22.8	77.4	66.6	62.1	90.7	87.0	79.3	77.3	87.4	30.0	78.2	75.5	75.3	59.2	84.1	4.9	56.0	92.5	21.5	15.7	86.6	34.7	100.0	60.2
gen_f1	89.9	28.9	84.4	47.3	28.8	58.4	90.5	89.0	88.0	84.8	90.7	89.8	45.3	90.1	87.5	82.5	30.0	86.5	90.0	84.9	84.0	50.6	50.2	87.5	89.1	34.7	88.8	91.8	35.5	44.0	28.7	82.5	78.7	76.9	93.6	92.7	86.1	84.1	91.5	44.3	85.9	83.9	85.5	62.2	89.2	42.9	58.8	94.3	41.4	47.5	89.7	45.8	99.6	71.3
gen_em	82.5	22.2	76.4	26.2	19.4	13.4	82.9	81.0	79.2	76.3	83.9	81.6	35.0	83.9	80.0	76.5	24.2	76.4	81.9	80.4	77.3	19.9	31.1	78.6	80.8	10.2	81.2	84.7	23.1	4.6	22.4	76.7	65.2	63.1	90.4	86.1	78.7	71.4	86.4	31.4	77.7	74.5	76.3	52.7	82.3	5.0	56.0	91.7	24.7	16.2	85.3	34.2	99.6	59.7
loc_f1	14.5	9.6	12.4	8.4	9.7	26.2	18.4	17.4	9.7	13.9	11.6	14.5	7.3	18.1	13.8	10.8	4.8	14.0	18.2	10.4	17.9	22.6	12.6	11.9	14.3	24.7	14.4	17.1	11.5	27.6	4.6	13.0	14.3	14.9	13.8	15.5	9.6	16.7	16.9	7.5	10.8	13.4	13.5	9.7	12.4	24.9	8.8	12.9	9.7	13.3	12.5	13.2	19.5	13.8
loc_em	9.8	5.5	7.5	3.7	6.3	1.5	12.5	12.5	6.3	8.9	7.7	9.7	4.5	12.6	8.8	6.8	3.2	8.3	11.9	7.3	11.7	2.2	5.2	7.8	9.1	1.7	10.0	12.0	5.7	1.9	1.8	8.8	8.3	10.2	9.5	11.2	6.1	10.7	11.7	5.0	7.1	8.9	8.3	5.6	6.0	0.8	5.4	8.3	5.2	6.3	8.2	7.1	13.7	7.4
port_f1	17.1	9.0	15.3	8.9	11.5	25.9	16.1	8.8	15.8	11.1	15.1	13.1	6.2	17.6	13.5	12.1	10.8	14.0	18.4	13.3	16.5	24.8	10.0	14.6	16.7	29.1	15.6	16.9	10.1	30.4	4.1	7.6	10.5	8.7	16.4	18.0	15.2	19.5	18.0	13.6	15.5	16.8	15.5	4.8	16.4	26.2	9.8	16.2	11.4	7.9	14.4	13.4	21.8	14.6
port_em	12.8	5.1	11.2	1.9	5.1	1.8	12.0	5.5	11.0	9.2	12.2	10.7	3.3	13.4	11.1	9.6	4.3	9.7	13.9	9.6	12.4	5.6	3.6	11.2	12.1	1.4	11.1	12.6	2.3	0.5	2.6	5.4	5.9	5.1	12.4	13.7	11.4	14.5	13.5	7.4	11.2	12.5	11.7	2.4	13.0	0.4	6.6	12.5	6.3	2.6	12.2	4.9	15.9	8.4

Table 11: Full Results on WFD.

D Detailed Results of Prompt-Based Fine-Tuning vs. Vanilla Fine-Tuning

We present the detailed results of few-shot training performance of Vanilla and TOPRO for all three tasks in Table 13 (Amazon Review), Table 14 (PAWS-X) and Table 15 (XNLI), as well as the T-test results for all tasks in few-shot conditions in Table 12.

Shot	Amazon		PAWS-X		XNLI	
	M	X	M	X	M	X
1	0.001	0.001	0.50	0.56*	0.01	0.12*
2	0.10	0.01	0.22*	0.08*	0.89*	0.18*
4	0.09*	0.02	0.80*	0.10*	0.05	0.07*
8	0.23*	0.04	0.83*	0.04	0.86*	0.14*
16	0.78*	0.11*	0.30*	0.05	0.27*	0.03
32	0.06*	0.16*	1.00*	0.58*	0.11*	0.01
64	0.03	0.18*	0.02	0.80*	0.09*	0.002
128	0.07*	0.11*	0.15*	0.82*	0.34*	0.01
256	0.73*	0.21*	0.12*	0.78*	0.07*	0.02
512	0.86*	0.01	0.04	0.90*	0.61*	0.004
1028	0.003	0.31*	0.03	0.55*	0.74*	0.03
full	0.005	0.40*	0.003	0.46*	0.005	0.44*

Table 12: T-Test results (p) for results of Vanilla and TOPRO in different few-shot conditions. M stands for mBERT and X stands for XLM-R. Insignificant results with a p value $>$ 0.05 are marked with *.

Shot	Model	en	de	es	fr	ja	zh	avg.
1	Vanilla-M	22.30	20.66	19.82	20.02	20.14	20.08	20.14
	ToPro -M	**28.52**	**26.05**	**26.98**	**26.18**	**25.96**	**25.01**	**26.04**
	Vanilla-X	21.98	22.15	21.69	21.79	21.42	21.52	21.71
	ToPro -X	**37.09**	**29.86**	**35.06**	**36.10**	**33.13**	**34.00**	**33.63**
2	Vanilla-M	24.37	23.14	23.00	22.70	21.27	21.36	22.29
	ToPro -M	**27.63**	**25.78**	**26.04**	**25.05**	**23.24**	**23.73**	**24.77**
	Vanilla-X	21.31	21.08	21.52	20.67	20.76	21.41	21.09
	ToPro -X	**35.63**	**31.82**	**33.46**	**34.40**	**33.35**	**32.70**	**33.14**
4	Vanilla-M	27.04	24.94	23.95	23.93	23.86	22.20	23.78
	ToPro -M	**30.63**	**26.87**	**27.67**	**26.34**	**25.44**	**26.05**	**26.47**
	Vanilla-X	29.74	29.96	29.67	30.87	26.12	28.89	29.10
	ToPro -X	**40.23**	**37.91**	**38.60**	**38.75**	**38.84**	**37.11**	**38.24**
8	Vanilla-M	29.95	26.82	26.75	26.91	24.18	25.70	26.07
	ToPro -M	**32.67**	**29.07**	**30.20**	**29.38**	**26.24**	**27.12**	**28.40**
	Vanilla-X	32.02	32.84	33.02	32.60	28.84	31.51	31.76
	ToPro -X	**42.23**	**35.63**	**40.55**	**39.79**	**39.65**	**38.33**	**38.79**
16	Vanilla-M	33.92	30.87	32.01	30.29	28.94	28.36	30.09
	ToPro -M	**35.27**	**31.66**	**32.10**	**31.37**	**29.70**	**28.58**	**30.68**
	Vanilla-X	38.97	39.42	38.70	38.84	34.61	35.72	37.45
	ToPro -X	**44.78**	**44.40**	**43.89**	**43.55**	**42.57**	**41.26**	**43.13**
32	Vanilla-M	36.73	31.26	31.64	31.69	28.94	29.08	30.52
	ToPro -M	**37.90**	**33.44**	**34.68**	**33.72**	**31.18**	**30.77**	**32.76**
	Vanilla-X	44.92	45.42	44.45	44.78	42.16	41.85	43.73
	ToPro -X	**47.51**	**47.12**	**46.67**	**45.78**	**44.24**	**42.70**	**45.30**
64	Vanilla-M	39.85	33.76	35.20	34.65	30.98	29.90	32.90
	ToPro -M	**41.62**	**36.25**	**37.84**	**36.15**	**32.97**	**32.56**	**35.15**
	Vanilla-X	48.06	**48.48**	46.77	**47.34**	44.01	42.05	45.73
	ToPro -X	**49.42**	48.16	**47.99**	46.93	**45.58**	**44.00**	**46.53**
128	Vanilla-M	43.29	35.52	37.50	36.38	32.36	31.51	34.65
	ToPro -M	**44.19**	**38.39**	**39.84**	**38.74**	**34.62**	**33.71**	**37.06**
	Vanilla-X	50.40	50.75	48.37	48.12	46.26	44.80	47.66
	ToPro -X	**50.75**	**51.24**	**49.75**	**49.22**	**47.39**	**45.35**	**48.59**
256	Vanilla-M	**45.64**	37.15	39.23	38.20	**33.54**	**32.86**	36.20
	ToPro -M	45.39	**37.71**	**39.99**	**40.31**	32.55	32.82	**36.68**
	Vanilla-X	51.21	50.92	47.15	47.85	46.01	44.23	47.23
	ToPro -X	**51.40**	**52.18**	**50.22**	**49.81**	**47.65**	**45.60**	**49.09**
512	Vanilla-M	**47.66**	**37.57**	39.90	39.16	**33.82**	**33.64**	36.82
	ToPro -M	47.64	37.48	**40.63**	**40.99**	32.76	33.40	**37.05**
	Vanilla-X	51.90	51.69	49.21	49.67	46.23	43.96	48.15
	ToPro -X	**52.94**	**52.79**	**50.21**	**50.06**	**48.16**	**45.82**	**49.41**
1024	Vanilla-M	49.26	38.47	41.24	39.88	33.52	33.79	37.38
	ToPro -M	**49.63**	**41.47**	**43.54**	**41.97**	**36.52**	**34.54**	**39.61**
	Vanilla-X	51.33	48.55	45.06	44.91	42.85	41.79	44.63
	ToPro -X	**54.55**	**53.15**	**51.98**	**51.18**	**47.98**	**46.08**	**50.07**
full	Vanilla-M	58.92	45.69	48.02	47.45	35.07	**38.63**	42.97
	ToPro -M	**59.05**	**46.66**	**49.30**	**48.38**	**37.31**	38.26	**43.98**
	Vanilla-X	59.61	**60.14**	55.24	55.66	51.93	**49.82**	54.56
	ToPro -X	**60.06**	59.60	**55.72**	**55.89**	**52.34**	49.75	**54.66**

Table 13: Few-shot performance on Amazon.

Shot	Model	en	de	es	fr	ja	ko	zh	avg.
1	Vanilla-M	**54.38**	53.29	54.22	54.25	53.37	54.01	53.20	53.72
	TOPRO -M	53.21	**54.18**	**54.44**	**54.34**	**55.31**	**54.35**	**53.80**	**54.40**
	Vanilla-X	**51.95**	**51.75**	**51.57**	**51.62**	**51.95**	**51.73**	**51.80**	**51.74**
	TOPRO -X	50.19	48.53	50.68	46.83	50.80	44.55	49.91	48.55
2	Vanilla-M	**53.54**	**53.60**	**53.81**	**54.18**	**54.43**	**54.54**	**53.77**	**54.06**
	TOPRO -M	52.38	53.04	53.34	53.13	54.35	53.90	51.82	53.26
	Vanilla-X	**54.95**	**54.73**	**54.30**	**54.57**	**54.25**	**54.05**	**54.32**	**54.37**
	TOPRO -X	51.59	50.25	51.65	48.86	51.31	46.30	50.70	49.85
4	Vanilla-M	**53.93**	**53.11**	53.38	**53.94**	53.85	**54.28**	**53.71**	**53.71**
	TOPRO -M	52.40	53.07	**53.64**	53.41	**54.79**	53.53	51.20	53.27
	Vanilla-X	53.15	**54.45**	**53.99**	**53.90**	**53.81**	**53.79**	**53.64**	**53.93**
	TOPRO -X	**53.54**	51.25	53.00	49.05	53.46	45.29	51.83	50.65
8	Vanilla-M	**54.30**	53.50	**53.51**	**54.02**	**54.03**	**53.94**	**54.15**	**53.86**
	TOPRO -M	52.81	**54.12**	53.42	53.31	53.98	53.51	51.93	53.38
	Vanilla-X	**54.60**	**55.13**	**54.68**	**54.80**	**55.46**	**55.10**	**55.14**	**55.05**
	TOPRO -X	53.18	52.65	53.03	51.22	52.48	48.83	52.21	51.74
16	Vanilla-M	**54.08**	50.86	52.04	52.66	51.77	52.27	51.23	51.81
	TOPRO -M	52.81	**53.08**	**53.80**	**53.20**	**53.51**	**53.95**	**52.09**	**53.27**
	Vanilla-X	**54.45**	**54.84**	**54.45**	**54.54**	**54.96**	**54.56**	**54.78**	**54.69**
	TOPRO -X	53.73	51.58	53.24	49.95	53.21	48.28	52.31	51.43
32	Vanilla-M	**54.03**	52.94	53.48	**53.65**	53.13	53.58	**53.08**	**53.31**
	TOPRO -M	52.99	**52.97**	**53.75**	53.14	**53.57**	**54.16**	51.42	53.17
	Vanilla-X	52.44	**53.95**	52.96	**53.21**	53.46	**54.05**	**53.94**	**53.60**
	TOPRO -X	**53.63**	51.96	**53.44**	50.51	**53.61**	49.84	52.73	52.01
64	Vanilla-M	**55.44**	**55.42**	**55.46**	**55.97**	**54.80**	**55.92**	**56.41**	**55.66**
	TOPRO -M	53.95	54.59	54.05	54.48	54.51	54.95	52.61	54.20
	Vanilla-X	55.20	**55.35**	54.69	**54.95**	**55.84**	**55.09**	**55.39**	**55.22**
	TOPRO -X	**56.60**	54.95	**55.90**	54.59	55.63	51.51	55.29	54.64
128	Vanilla-M	**56.63**	**56.29**	**56.69**	**56.43**	55.31	55.70	**55.75**	**56.03**
	TOPRO -M	55.54	55.76	55.28	55.26	**55.88**	**55.75**	55.61	55.59
	Vanilla-X	54.61	54.99	54.44	54.80	55.24	**55.14**	54.98	54.93
	TOPRO -X	**58.66**	**56.28**	**57.95**	**54.91**	**56.09**	52.39	**57.35**	**55.83**
256	Vanilla-M	58.66	56.00	56.38	56.93	55.36	55.77	55.65	56.02
	TOPRO -M	**61.84**	**60.51**	**60.65**	**60.90**	**58.56**	**58.70**	**59.70**	**59.84**
	Vanilla-X	59.30	**58.23**	58.79	**58.54**	57.18	**57.54**	**57.70**	**57.99**
	TOPRO -X	**59.94**	57.75	**59.58**	57.86	**57.28**	54.31	57.35	57.35
512	Vanilla-M	64.23	59.38	60.00	60.15	56.90	56.84	56.79	58.34
	TOPRO -M	**73.47**	**69.74**	**70.23**	**70.20**	**63.84**	**64.56**	**66.97**	**67.59**
	Vanilla-X	**77.03**	**71.28**	72.09	**72.46**	**63.43**	**63.79**	66.53	**68.26**
	TOPRO -X	76.94	71.01	**72.29**	71.24	63.19	63.28	**66.61**	67.94
1024	Vanilla-M	74.43	68.44	69.47	70.01	61.95	61.13	64.69	65.95
	TOPRO -M	**81.06**	**74.58**	**76.08**	**76.15**	**66.05**	**66.76**	**70.64**	**71.71**
	Vanilla-X	86.33	**79.23**	80.86	**80.74**	**69.25**	**68.18**	**73.26**	**75.25**
	TOPRO -X	**87.84**	78.94	**81.53**	80.58	67.68	68.01	71.85	74.76
full	Vanilla-M	93.85	84.94	87.11	86.55	73.39	72.44	77.01	80.24
	TOPRO -M	**94.21**	**86.06**	**88.17**	**87.91**	**75.79**	**75.82**	**79.22**	**82.16**
	Vanilla-X	94.33	86.92	88.55	**89.04**	**76.07**	74.71	79.75	82.51
	TOPRO -X	**94.90**	**87.06**	**88.87**	88.86	75.53	**75.40**	**80.63**	**82.73**

Table 14: Few-shot performance on PAWS-X.

Shot	Model	en	ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh	avg.
1	Vanilla-M	33.58	32.97	32.97	33.46	32.70	33.33	33.43	32.44	32.93	32.85	33.12	33.05	32.96	33.00	32.99	33.02
	TOPRO -M	**37.58**	**34.93**	**33.56**	**35.95**	**35.02**	**34.25**	**36.38**	**33.93**	**36.76**	**34.62**	**33.83**	**34.07**	**34.22**	**36.43**	**37.41**	**35.10**
	Vanilla-X	33.73	33.07	32.86	33.51	32.66	33.40	33.54	32.50	33.04	33.15	33.18	33.14	33.00	33.08	33.04	33.08
	TOPRO -X	**39.26**	**34.61**	**34.85**	**36.28**	**36.88**	**33.59**	**34.92**	**39.76**	**34.47**	**36.53**	**36.33**	**36.56**	**37.03**	**37.40**	**36.61**	**36.13**
2	Vanilla-M	34.67	**34.98**	36.21	**36.15**	**35.46**	**36.91**	**36.42**	34.34	35.42	**34.67**	**34.20**	**35.40**	34.04	36.18	35.61	**35.43**
	TOPRO -M	**38.38**	34.85	**34.02**	35.07	35.20	33.44	35.70	**35.63**	**35.65**	34.50	33.78	34.35	**34.67**	**36.57**	**37.12**	35.04
	Vanilla-X	34.84	34.33	35.51	35.62	34.99	**36.25**	35.86	34.14	35.09	34.39	33.95	34.87	33.76	35.55	35.02	34.95
	TOPRO -X	**39.22**	**36.54**	**36.73**	**38.48**	**37.83**	34.21	**37.91**	**38.87**	**35.56**	**37.16**	**38.42**	**38.01**	**37.75**	**38.25**	**36.98**	**37.34**
4	Vanilla-M	37.91	**35.47**	**36.12**	36.20	35.03	**36.22**	36.09	34.60	35.60	**35.01**	**34.35**	**35.49**	**34.49**	36.28	35.74	35.48
	TOPRO -M	**38.04**	35.43	34.64	**36.67**	**36.50**	33.66	**36.63**	**36.07**	**36.83**	34.87	33.42	35.41	34.44	**37.06**	**37.07**	**35.62**
	Vanilla-X	37.55	34.31	35.08	35.11	34.09	**35.06**	34.85	33.74	**34.53**	34.09	33.58	34.39	33.71	35.09	34.56	34.44
	TOPRO -X	**38.79**	**36.03**	**35.23**	**37.49**	**37.36**	33.50	**36.54**	**38.79**	34.21	**37.11**	**37.79**	**36.47**	**37.58**	**37.96**	**36.22**	**36.59**
8	Vanilla-M	**40.83**	**37.39**	**38.56**	**38.69**	**37.77**	**39.25**	**39.06**	36.38	37.72	**37.54**	**36.46**	**38.07**	**36.28**	**38.22**	37.76	**37.80**
	TOPRO -M	38.71	36.59	35.73	37.20	37.33	34.88	38.05	**38.22**	**38.32**	35.37	35.40	36.48	35.99	38.20	**38.93**	36.91
	Vanilla-X	40.84	36.52	37.57	37.97	36.85	**38.50**	**38.35**	35.70	37.00	36.77	35.57	37.33	35.57	37.56	36.95	37.01
	TOPRO -X	**41.58**	**37.81**	**37.61**	**39.74**	**39.06**	35.07	37.65	**39.78**	**37.26**	**38.64**	**40.32**	**38.79**	**38.65**	**40.33**	**38.54**	**38.52**
16	Vanilla-M	42.42	39.56	40.71	40.36	39.63	**41.49**	41.14	37.86	39.60	**38.27**	37.35	38.77	37.44	40.76	40.25	39.51
	TOPRO -M	**44.52**	**42.10**	**41.96**	**40.85**	**42.18**	40.63	**43.98**	**41.17**	**43.10**	36.50	**38.83**	**41.71**	**38.95**	**43.40**	**43.14**	**41.32**
	Vanilla-X	42.65	39.37	40.33	40.09	39.15	**41.12**	40.73	37.72	39.44	38.02	37.34	38.63	37.19	40.73	40.01	39.28
	TOPRO -X	**49.72**	**42.15**	**43.51**	**47.38**	**46.22**	40.19	**44.09**	**45.59**	**43.14**	**44.81**	**46.16**	**45.39**	**44.43**	**47.35**	**45.69**	**44.72**
32	Vanilla-M	46.18	40.39	41.17	41.25	40.39	42.65	41.88	38.69	40.77	**38.29**	38.47	39.62	38.82	41.18	40.89	40.32
	TOPRO -M	**49.02**	**45.64**	**46.01**	**44.64**	**47.57**	**45.00**	**48.32**	**45.06**	**46.37**	38.28	**43.39**	**43.68**	**43.88**	**47.18**	**47.78**	**45.20**
	Vanilla-X	46.11	39.69	40.44	40.57	39.81	42.05	41.28	38.30	40.25	37.71	37.99	39.05	38.17	40.27	40.00	39.68
	TOPRO -X	**52.27**	**46.87**	**48.41**	**49.79**	**49.12**	**45.55**	**48.85**	**48.42**	**48.10**	**45.90**	**49.20**	**47.88**	**46.58**	**49.84**	**48.55**	**48.08**
64	Vanilla-M	52.10	45.26	46.64	48.10	46.32	49.44	48.57	42.71	45.45	39.13	40.24	42.19	42.41	47.23	46.91	45.04
	TOPRO -M	**55.04**	**50.28**	**51.76**	**52.60**	**52.90**	**50.46**	**53.85**	**49.57**	**51.68**	**42.26**	**46.38**	**49.01**	**48.85**	**52.89**	**52.57**	**50.36**
	Vanilla-X	51.86	44.99	46.39	47.86	45.84	48.92	48.47	42.99	45.25	39.04	40.35	42.43	42.51	47.08	46.70	44.92
	TOPRO -X	**59.35**	**50.75**	**53.38**	**55.47**	**55.32**	**50.92**	**55.71**	**53.11**	**52.67**	**51.31**	**53.99**	**52.95**	**51.30**	**55.51**	**54.41**	**53.34**
128	Vanilla-M	58.61	51.91	54.23	54.89	54.32	**56.27**	55.30	49.05	52.87	43.18	46.02	49.56	48.28	54.02	54.06	51.71
	TOPRO -M	**59.12**	**53.87**	**55.09**	**56.44**	**55.33**	55.00	**56.09**	**52.36**	**54.71**	**45.25**	**49.41**	**52.44**	**51.35**	**55.62**	**55.98**	**53.50**
	Vanilla-X	58.27	51.41	53.86	54.61	53.85	55.90	54.89	48.68	52.21	42.87	46.23	49.26	47.89	53.55	53.90	51.36
	TOPRO -X	**64.78**	**56.50**	**60.23**	**60.77**	**60.55**	**59.51**	**61.20**	**57.41**	**59.13**	**55.12**	**58.44**	**58.15**	**55.36**	**60.24**	**59.68**	**58.73**
256	Vanilla-M	61.88	53.54	56.61	57.25	56.20	**58.77**	57.91	51.31	55.45	44.97	46.97	52.75	50.07	56.51	56.76	53.94
	TOPRO -M	**62.30**	**54.82**	**56.96**	**57.92**	**56.48**	58.69	**58.39**	**53.58**	**57.09**	**45.55**	**49.06**	**53.64**	**52.41**	**57.81**	**58.06**	**55.03**
	Vanilla-X	61.68	53.30	56.19	57.01	55.91	58.47	57.74	51.13	55.22	44.86	46.68	52.77	49.79	56.24	56.33	53.69
	TOPRO -X	**66.55**	**58.08**	**62.26**	**62.24**	**61.23**	**62.88**	**63.44**	**58.56**	**60.42**	**54.77**	**59.95**	**59.95**	**56.59**	**62.28**	**61.18**	**60.27**
512	Vanilla-M	64.94	56.75	59.66	60.73	58.53	**61.99**	60.89	53.69	58.94	46.24	48.58	55.50	52.56	59.71	59.89	56.69
	TOPRO -M	**65.39**	**57.36**	**60.18**	**61.03**	**58.95**	61.59	**61.04**	**55.07**	**59.52**	**47.23**	**50.48**	**55.98**	**54.08**	**60.25**	**60.41**	**57.37**
	Vanilla-X	64.92	56.33	59.53	60.47	58.11	61.92	60.59	53.36	58.53	45.92	47.99	55.25	52.15	59.32	59.49	56.35
	TOPRO -X	**70.13**	**61.99**	**66.33**	**65.47**	**64.91**	**67.43**	**66.72**	**60.53**	**64.80**	**57.27**	**63.16**	**63.35**	**58.78**	**65.31**	**64.74**	**63.63**
1024	Vanilla-M	65.90	56.85	59.73	61.10	58.40	**62.73**	**62.07**	54.57	59.38	46.46	48.46	56.19	**54.21**	60.32	60.51	57.21
	TOPRO -M	**66.77**	**57.83**	**59.94**	**61.53**	**59.42**	62.05	61.99	**55.37**	**59.54**	**47.44**	**49.10**	**56.40**	53.91	**60.48**	**60.62**	**57.54**
	Vanilla-X	65.67	56.88	59.61	60.95	57.99	62.47	61.93	54.48	59.30	46.36	48.21	56.01	54.29	60.15	60.25	57.06
	TOPRO -X	**71.51**	**63.04**	**67.62**	**66.26**	**66.27**	**68.64**	**67.72**	**62.02**	**65.86**	**58.12**	**64.33**	**64.41**	**60.46**	**66.36**	**65.50**	**64.76**
full	Vanilla-M	82.57	65.12	68.97	71.40	66.30	74.22	73.68	60.02	68.95	50.24	53.15	62.02	57.96	69.80	68.91	65.05
	TOPRO -M	82.57	**65.55**	**69.47**	**71.57**	**67.43**	**75.10**	**74.57**	**60.57**	**69.55**	**51.13**	**54.58**	**62.64**	**58.04**	**70.74**	**70.08**	**65.79**
	Vanilla-X	84.91	**71.86**	77.78	76.86	75.96	79.25	78.21	69.92	**75.79**	**65.21**	72.02	73.12	66.07	74.71	73.72	73.61
	TOPRO -X	**84.97**	71.81	**77.92**	**77.35**	**76.11**	**79.31**	**78.75**	**70.10**	75.43	65.13	**72.39**	**73.23**	**66.95**	**75.05**	**73.92**	**73.82**

Table 15: Few-shot performance on XNLI.

E Detailed Results for TOPRO

We present the detailed results of the cross-lingual evaluation performance of Vanilla, Prompt Tuning, and TOPRO in Table 16 (PAN-X) and Table 17 (UDPOS).

lang.	en	af	ar	az	bg	bn	de	el	es	et	eu	fa	fi
B (Vanilla)	83.83	78.07	44.09	67.58	78.31	70.09	79.10	71.85	73.95	77.96	65.44	42.43	78.74
B (PT)	79.09	71.37	39.52	63.47	73.28	58.81	74.14	63.35	68.05	73.84	61.00	34.86	74.01
B (ToPro)	**92.80**	**90.87**	**62.62**	**85.30**	**89.61**	**78.33**	**92.40**	**89.88**	**84.94**	**90.07**	**85.35**	**69.52**	**91.25**
X (Vanilla)	81.31	75.03	47.26	61.37	77.02	68.97	74.07	74.93	70.51	70.73	58.07	48.73	75.44
X (PT)	75.94	69.92	43.75	58.57	72.15	53.42	68.09	64.12	65.21	65.43	47.97	38.65	70.31
X (ToPro)	**92.21**	**90.02**	**67.84**	**84.02**	**88.20**	**72.06**	**91.22**	**91.22**	**83.63**	**88.26**	**84.59**	**62.82**	**90.72**
T (Vanilla)	77.14	76.94	49.99	62.00	72.98	60.32	76.19	76.88	67.81	74.25	67.12	40.46	75.93
T (ToPro)	**96.52**	**96.76**	**89.13**	**94.78**	**96.11**	**90.74**	**97.21**	**96.22**	**93.90**	**95.80**	**94.62**	**87.93**	**96.71**

lang.	fr	gu	he	hi	hu	id	it	ja	jv	ka	kk	ko	lt
B (Vanilla)	80.40	53.89	55.80	68.17	76.16	61.21	81.10	28.25	61.58	67.94	47.21	61.60	74.41
B (PT)	75.02	32.07	52.00	62.38	70.88	58.39	78.11	23.76	57.23	61.45	46.06	58.51	69.86
B (ToPro)	**87.15**	**87.22**	**83.27**	**80.88**	**90.91**	**77.99**	**91.24**	**69.29**	**80.28**	**87.25**	**80.95**	**83.94**	**87.99**
X (Vanilla)	75.81	57.12	51.54	68.11	76.42	48.04	77.58	19.26	57.86	67.02	40.79	50.36	73.85
X (PT)	69.14	47.54	43.64	60.58	70.17	45.33	71.55	16.98	41.49	57.22	40.66	44.73	67.08
X (ToPro)	**86.20**	**88.11**	**82.49**	**79.28**	**91.38**	**69.35**	**89.36**	**66.87**	**74.29**	**87.50**	**83.14**	**81.78**	**88.09**
T (Vanilla)	73.68	64.18	68.83	61.90	74.01	64.28	77.33	46.19	67.79	70.17	65.10	60.24	72.09
T (ToPro)	**94.55**	**96.17**	**92.93**	**92.69**	**96.98**	**91.22**	**96.35**	**89.71**	**90.59**	**96.02**	**93.73**	**93.40**	**95.57**

lang.	ml	mr	ms	my	nl	pa	pl	pt	qu	ro	ru	sw	ta
B (Vanilla)	56.00	57.77	67.05	53.36	82.23	34.29	80.74	79.77	64.53	73.97	65.33	70.08	53.33
B (PT)	50.35	51.17	63.17	43.18	77.78	31.36	77.38	74.00	46.06	59.57	58.14	60.57	49.08
B (ToPro)	**82.57**	**82.93**	**81.55**	**82.65**	**92.35**	**59.67**	**90.87**	**87.25**	**77.50**	**81.88**	**84.71**	**79.33**	**77.81**
X (Vanilla)	59.85	60.74	66.13	53.41	79.67	50.31	77.64	76.83	60.49	70.45	62.54	69.51	54.62
X (PT)	51.08	48.43	45.86	44.94	74.88	33.83	73.04	70.12	45.36	59.48	54.84	57.57	47.83
X (ToPro)	**85.55**	**81.75**	**74.39**	**85.10**	**92.00**	**69.72**	**90.66**	**85.99**	**77.57**	**83.60**	**80.65**	**77.32**	**81.30**
T (Vanilla)	62.21	61.71	68.06	44.70	77.43	53.71	75.31	70.83	62.18	69.10	66.16	66.60	62.69
T (ToPro)	**94.77**	**93.42**	**85.70**	**93.66**	**96.93**	**86.18**	**96.34**	**94.81**	**87.35**	**94.16**	**94.07**	**91.90**	**92.52**

lang.	te	th	tl	tr	uk	ur	vi	yo	zh	avg.
B (Vanilla)	50.86	0.77	71.14	74.66	71.30	33.22	69.69	49.29	43.51	62.73
B (PT)	47.77	0.54	71.54	67.16	65.20	26.49	67.17	37.71	40.73	56.76
B (ToPro)	**83.83**	**68.37**	**82.54**	**87.29**	**85.94**	**63.18**	**86.04**	**64.70**	**68.39**	**81.91**
X (Vanilla)	48.20	3.09	69.84	75.58	73.43	59.48	67.92	50.25	25.28	61.30
X (PT)	40.89	3.67	62.14	64.48	61.21	38.17	61.68	35.57	24.51	53.05
X (ToPro)	**84.73**	**19.56**	**78.35**	**89.35**	**85.74**	**61.11**	**82.18**	**66.38**	**66.09**	**80.03**
T (Vanilla)	66.67	29.23	63.28	69.28	69.94	37.75	61.28	61.24	50.87	64.19
T (ToPro)	**94.82**	**79.33**	**90.34**	**96.21**	**93.45**	**89.06**	**92.94**	**84.54**	**90.37**	**92.82**

Table 16: Detailed results of the cross-lingual evaluation on PAN-X.

lang.	en	af	ar	bg	de	el	es	et	eu	fa	fi	fr	he
B (Vanilla)	95.28	86.10	53.51	85.65	86.36	81.92	**86.79**	80.78	58.75	66.10	80.33	**84.59**	56.27
B (PT)	94.96	86.06	55.59	85.81	86.03	80.87	85.04	76.74	59.99	66.91	77.75	79.67	56.12
B (ToPro)	**95.82**	**89.37**	**70.02**	**88.45**	**89.46**	**85.72**	85.93	**84.64**	**68.86**	**68.33**	**82.96**	84.43	**80.68**
X (Vanilla)	95.64	87.88	65.41	88.48	88.03	86.63	**88.31**	85.96	70.07	69.22	85.32	**86.57**	66.38
X (PT)	95.18	87.92	65.69	88.35	87.76	**86.78**	87.98	84.96	66.71	68.57	84.60	86.21	66.12
X (ToPro)	**96.05**	**89.88**	**70.06**	**89.04**	**89.61**	86.14	87.08	**86.90**	**71.95**	**70.04**	**85.80**	81.21	**80.50**
T (Vanilla)	89.67	85.02	63.56	78.38	79.86	75.44	83.99	78.35	68.49	66.47	77.50	82.10	64.19
T (ToPro)	**97.57**	**92.18**	**78.79**	**92.72**	**92.35**	**88.50**	**89.72**	**89.93**	**82.63**	**81.59**	**89.18**	**90.51**	**87.87**

lang.	hi	hu	id	it	ja	kk	ko	lt	mr	nl	pl	pt	ro
B (Vanilla)	63.54	78.92	71.67	**88.46**	47.05	70.53	50.82	79.79	70.35	89.00	81.77	**86.44**	78.00
B (PT)	64.54	78.40	71.47	86.78	46.77	70.19	51.63	76.93	67.24	88.49	81.15	86.02	77.14
B (ToPro)	**73.16**	**79.48**	**76.30**	86.45	**52.10**	**74.98**	**64.68**	**83.54**	**75.75**	**89.50**	**84.97**	85.36	**81.15**
X (Vanilla)	69.35	**82.97**	72.83	87.79	25.62	76.14	52.75	84.67	82.61	89.26	83.91	**87.16**	84.23
X (PT)	69.19	82.72	72.50	**88.88**	22.17	74.93	53.29	83.11	81.22	88.95	84.24	87.11	83.80
X (ToPro)	**72.98**	80.90	**76.93**	86.53	**54.78**	**76.61**	**64.14**	**87.16**	**80.09**	**89.54**	**85.74**	86.37	**85.69**
T (Vanilla)	69.21	76.85	72.11	82.71	50.81	71.57	51.22	76.92	72.58	83.85	77.39	82.78	74.51
T (ToPro)	**87.89**	**90.75**	**85.92**	**90.99**	**78.12**	**87.18**	**76.79**	**89.73**	**89.76**	**93.01**	**91.16**	**90.74**	**88.58**

lang.	ru	ta	te	th	tl	tr	uk	ur	vi	wo	yo	zh	avg.
B (Vanilla)	85.82	58.78	**76.63**	41.08	**82.30**	69.46	81.04	55.04	55.98	30.93	59.56	62.62	70.89
B (PT)	86.58	59.33	74.25	37.00	77.59	66.17	81.32	56.01	54.69	29.19	57.50	63.88	69.91
B (ToPro)	**90.02**	**72.21**	75.70	**56.92**	81.71	**71.29**	**86.95**	**67.08**	**58.77**	**33.38**	**65.29**	**72.17**	**76.16**
X (Vanilla)	89.16	61.94	**84.38**	44.73	86.80	**74.22**	85.22	58.88	58.48	**30.07**	**26.12**	32.08	72.42
X (PT)	88.50	61.91	82.11	40.80	**88.64**	72.74	84.85	60.68	57.34	28.79	25.07	33.81	71.86
X (ToPro)	**90.70**	**72.78**	83.79	**70.01**	82.11	73.85	**87.17**	**66.82**	**59.79**	19.38	19.53	**76.38**	**76.16**
T (Vanilla)	82.12	62.85	78.65	64.06	73.73	68.58	77.17	64.63	58.43	**54.89**	66.74	43.90	71.39
T (ToPro)	**93.48**	**83.23**	**90.65**	**79.67**	**93.11**	**85.46**	**90.67**	**85.10**	**79.03**	54.01	**72.71**	**82.43**	**86.11**

Table 17: Detailed results of the cross-lingual evaluation on UDPOS.

F Full Results of GNNAVI

The complete results are provided in Table 18. Each value in the table represents the average accuracy over five experiments conducted with different random seeds.

k	Method	#Param	SST-2	EmoC	TREC	Amazon	AGNews	Average	#Param	SST-2	EmoC	TREC	Amazon	AGNews	Average
		GPT2-XL							**Llama2**						
0	ICL	-	55.44	6.48	54.68	53.32	72.12	48.41	-	67.55	9.60	70.36	94.98	84.14	65.33
5	ICL	-	63.17	6.30	57.68	53.67	50.43	46.25	-	86.93	20.18	45.72	92.30	80.16	65.06
	LoRA	2.5M	91.98	50.60	75.20	88.80	**85.20**	78.36	4.2M	**95.42**	64.20	**88.40**	91.80	86.60	85.28
	Prefix	6.1M	59.13	73.46	32.92	60.00	75.40	60.18	39.3M	50.96	58.56	21.36	49.36	25.78	41.20
	Adapter	15.4M	79.82	76.00	**79.60**	**91.45**	81.25	81.62	198M	50.92	**84.05**	18.80	49.45	24.80	45.60
	FPFT	1.6B	62.13	61.30	65.28	73.00	80.82	68.51	6.7B	94.63	61.92	81.72	**95.86**	**87.58**	84.34
	GNNAVI-GCN	2.6M	**84.31**	75.48	76.72	90.90	83.16	**82.11**	16.8M	94.56	78.30	83.2	94.00	86.25	86.63
	GNNAVI-SAGE	5.1M	81.95	**78.70**	77.92	88.66	82.88	82.02	33.6M	92.91	80.12	80.80	95.66	86.06	**87.11**
10	LoRA	2.5M	**88.08**	53.20	86.40	90.60	86.80	81.02	4.2M	**94.73**	63.00	**92.80**	92.60	**90.40**	86.71
	Prefix	6.1M	51.08	77.58	38.16	65.94	61.48	58.85	39.3M	50.80	**76.98**	21.20	51.42	26.44	45.37
	Adapter	15.4M	86.70	70.65	**87.40**	90.60	86.15	84.30	198M	50.92	85.60	41.00	52.20	52.15	56.37
	FPFT	1.6B	69.01	71.90	52.48	75.82	81.34	70.11	6.7B	92.91	68.06	84.24	96.22	88.64	86.01
	GNNAVI-GCN	2.6M	84.63	**83.97**	74.80	91.57	**87.00**	**84.39**	16.8M	91.86	70.75	82.40	**96.35**	89.30	84.99
	GNNAVI-SAGE	5.1M	87.41	77.98	78.28	**91.90**	84.52	84.02	33.6M	94.06	76.02	83.96	95.76	87.64	**87.49**
20	LoRA	2.5M	85.09	69.00	86.00	**94.00**	**89.20**	84.66	4.2M	95.64	70.80	83.60	**96.20**	**90.60**	87.37
	Prefix	6.1M	56.68	**83.28**	39.20	61.22	80.62	64.20	39.3M	50.57	78.70	27.92	52.08	26.30	47.11
	Adapter	15.4M	88.42	74.65	**89.00**	89.45	86.50	85.60	198M	50.92	**85.80**	18.80	56.40	24.80	47.34
	FPFT	1.6B	73.10	70.72	68.36	77.40	80.44	74.00	6.7B	**95.32**	69.96	**88.08**	95.52	89.04	87.58
	GNNAVI-GCN	2.6M	86.93	76.23	79.67	92.70	86.07	84.32	16.8M	94.78	75.25	84.80	96.00	89.30	88.27
	GNNAVI-SAGE	5.1M	**88.67**	78.96	82.52	92.02	86.24	**85.68**	33.6M	94.56	79.92	84.56	95.64	88.54	**88.64**
50	LoRA	2.5M	89.45	74.80	54.80	**93.60**	**91.80**	80.89	4.2M	93.12	72.40	**94.40**	95.40	**91.60**	89.20
	Prefix	6.1M	50.90	**79.78**	26.72	74.42	74.40	61.24	39.3M	50.48	76.22	28.08	50.96	27.60	46.67
	Adapter	15.4M	86.75	77.85	**91.60**	90.50	88.75	87.09	198M	50.92	76.80	44.40	49.45	33.45	51.00
	FPFT	1.6B	70.60	71.68	76.40	67.84	83.10	73.92	6.7B	**95.46**	74.20	91.92	95.82	90.48	89.58
	GNNAVI-GCN	2.6M	89.49	79.50	87.93	92.40	87.43	**87.35**	16.8M	95.07	**83.05**	88.70	95.85	90.80	**90.81**
	GNNAVI-SAGE	5.1M	**90.14**	75.70	87.96	93.26	87.30	86.87	33.6M	94.72	79.04	90.72	**96.00**	90.68	90.23
100	LoRA	2.5M	89.22	**84.00**	88.40	93.20	84.80	87.92	4.2M	92.66	**86.60**	94.80	95.40	67.60	87.41
	Prefix	6.1M	56.26	72.28	32.04	69.48	51.18	56.25	39.3M	49.11	76.20	40.28	52.38	26.82	48.96
	Adapter	15.4M	86.93	82.85	**92.00**	92.40	87.60	88.36	198M	58.83	84.95	84.00	68.10	24.80	64.14
	FPFT	1.6B	72.82	73.42	68.56	78.74	84.86	75.68	6.7B	**95.07**	76.06	**96.20**	**96.20**	**91.04**	**90.91**
	GNNAVI-GCN	2.6M	89.41	81.30	90.20	92.67	87.97	88.31	16.8M	94.27	81.20	91.60	96.00	90.80	90.77
	GNNAVI-SAGE	5.1M	**90.46**	80.16	91.12	**93.28**	**88.58**	**88.72**	33.6M	94.45	81.20	90.88	96.08	90.78	90.68
200	LoRA	2.5M	**90.83**	80.80	90.80	82.00	86.20	86.13	4.2M	91.29	**86.80**	93.60	95.80	90.40	91.32
	Prefix	6.1M	50.92	80.18	69.80	59.80	79.08	67.96	39.3M	48.35	81.72	45.68	52.28	27.54	51.11
	Adapter	15.4M	88.65	80.70	**96.60**	92.30	**89.80**	89.61	198M	50.92	85.05	88.20	49.45	81.50	67.57
	FPFT	1.6B	68.97	73.70	80.16	74.82	85.34	76.60	6.7B	**95.64**	79.90	**96.76**	96.12	**91.44**	91.97
	GNNAVI-GCN	2.6M	90.67	78.82	91.88	92.94	89.20	88.70	16.8M	95.36	82.85	95.50	**96.45**	91.05	**92.24**
	GNNAVI-SAGE	5.1M	90.46	**82.68**	92.32	**93.44**	89.28	**89.64**	33.6M	95.30	81.94	94.76	95.96	90.68	91.73

Table 18: Results with different training methods (accuracy). k denotes the number of training examples per class. #Param denotes the number of trainable parameters. The best scores under the same circumstances of training examples are highlighted with **bold**.

G Detailed Results of Minimal Pair Probing

Raw results for English, Chinese, and German can be found in Figure 4, 5, and 6.

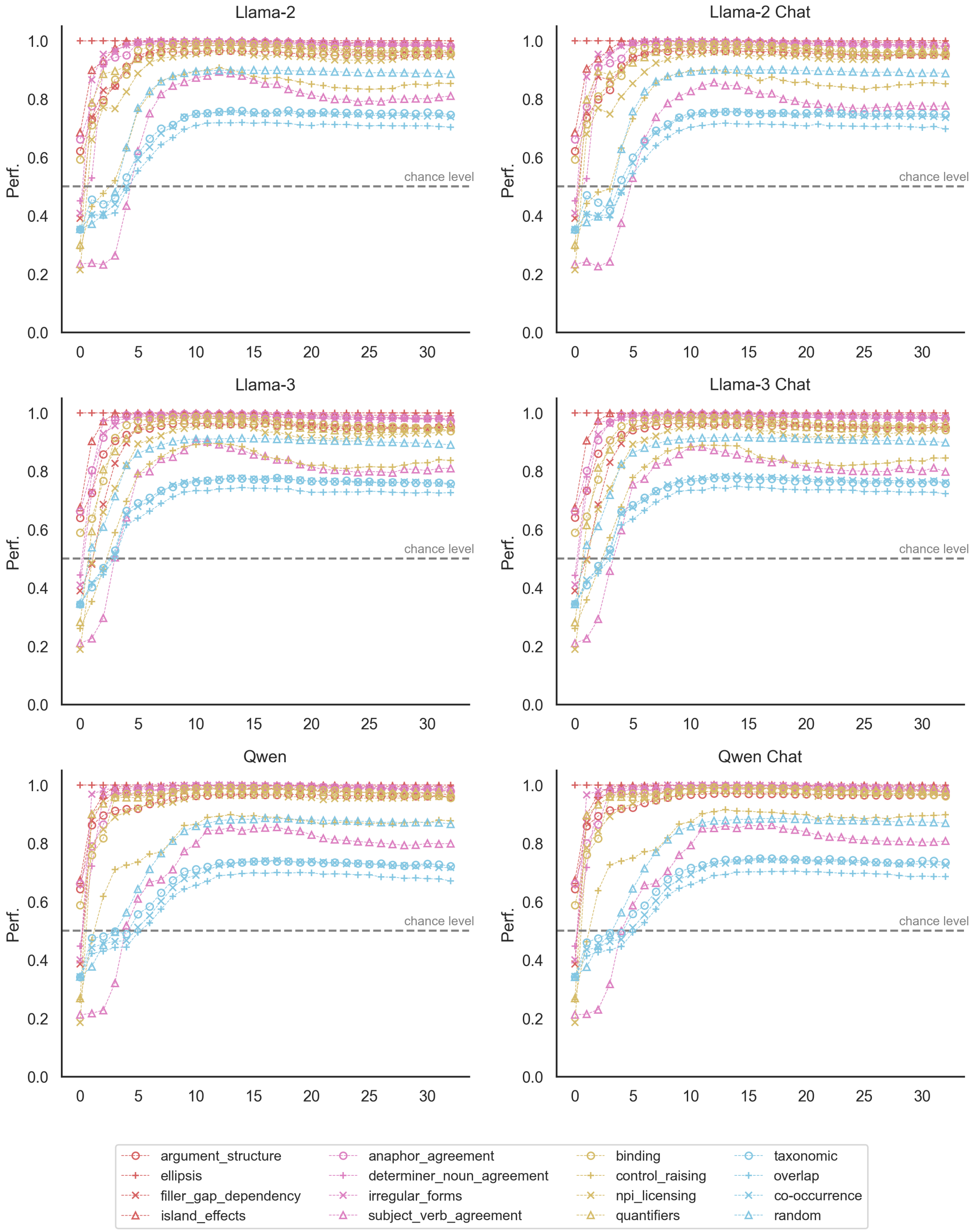


Figure 4: Detailed English decoding results on 6 models.

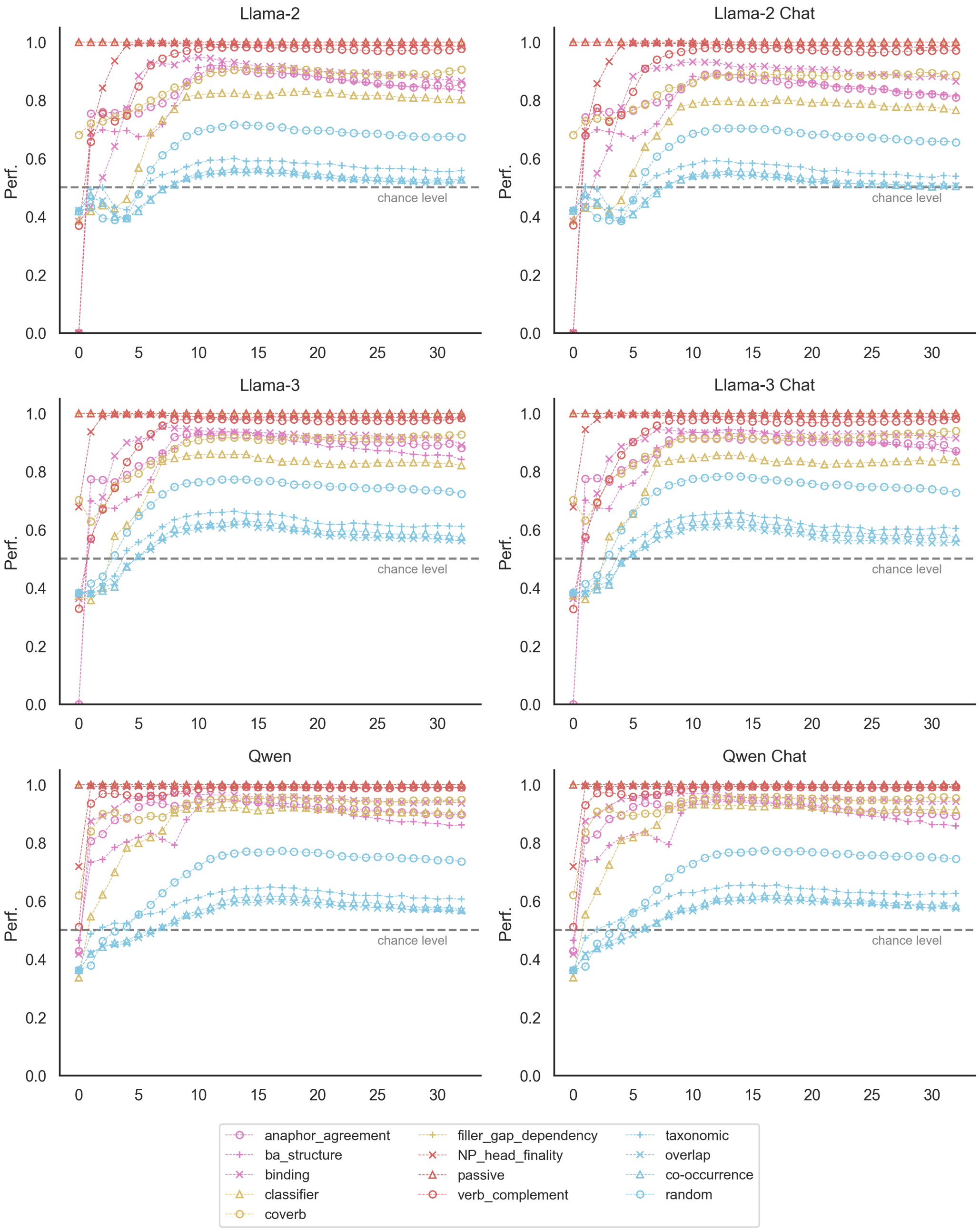


Figure 5: Detailed Chinese decoding results on 6 models. Notice that the pink, orange, and blue curves don't denote morphology or semantics as those in English do. They are made just to make it easier to distinguish in the figure. All non-red curves represent grammatical tasks and red curves represent conceptual tasks.

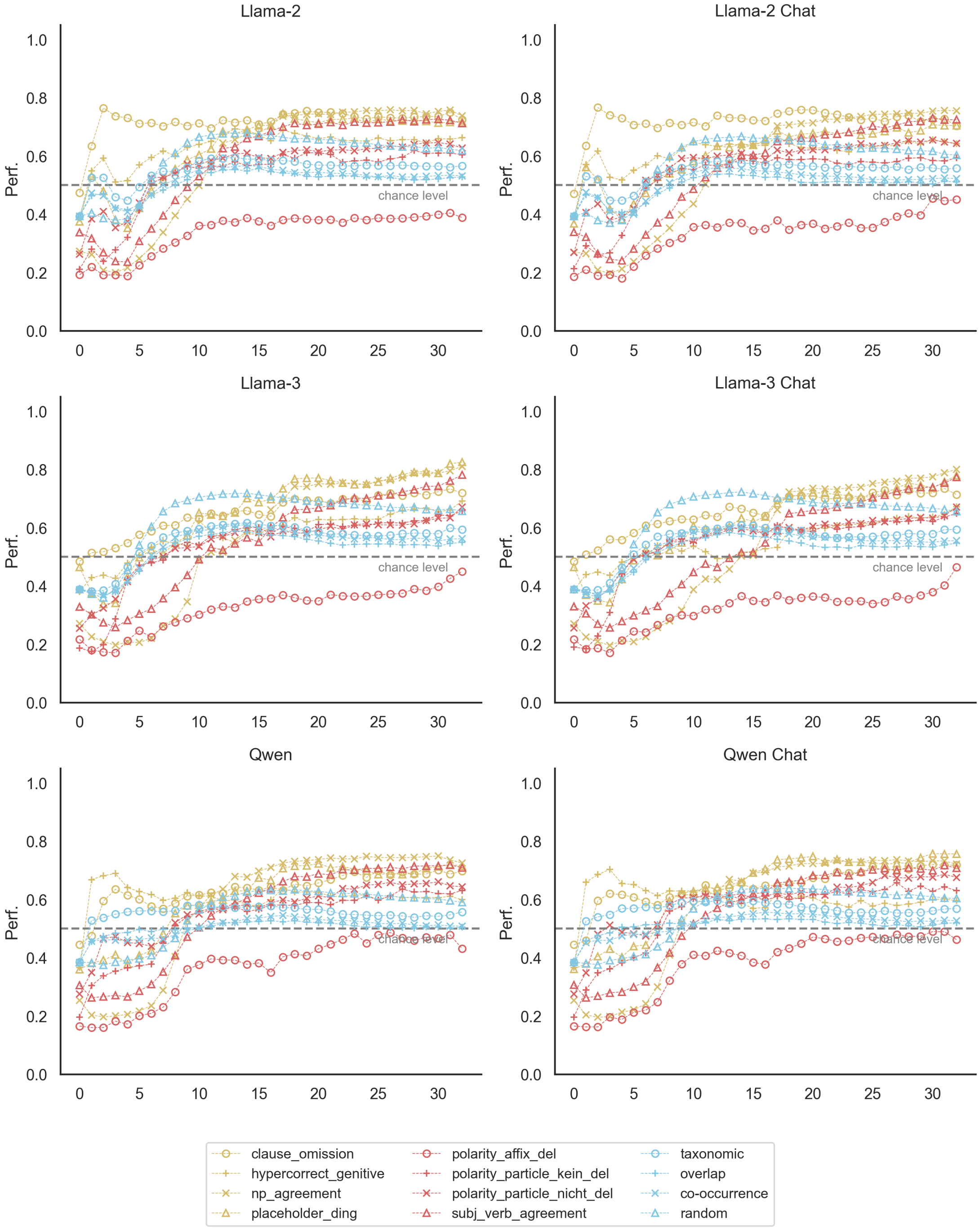


Figure 6: Detailed German decoding results on 6 models. All non-red curves are grammatical tasks, and red curves are conceptual tasks.

H Full Results and Detailed Experimental Setup of the Language Confusion Study

H.1 Full Experimental Results

Table 19 presents the full benchmarking results. Table 20 shows the full results of the CP replacement experiment. Tables 21 and 22 present the full results of robustness and generalization experiments.

H.2 Detailed Experimental Setup

H.2.1 Models

We primarily use three variants of the Llama3 family for our experiments:

- **Llama3-8B**: The baseline English-centric model without multilingual instruction tuning.
- **Llama3-8B-multilingual**: The multilingual instruction-tuned version, as described in (Devine, 2024).
- **Llama3.1-8B**: An improved model optimized for multilingual dialogue.

All models are used in their publicly released forms unless otherwise stated. For neuron editing experiments, we intervene on *Llama3-8B* using the strategies described in Section 5.

H.2.2 Datasets and Tasks

Language Confusion Benchmarking and Replacement Experiments We use the Language Confusion Benchmark (LCB) (Marchisio et al., 2024) for all language confusion detection and mitigation experiments. LCB covers 15 typologically diverse languages and comprises several monolingual and cross-lingual datasets:

- **Monolingual sources**: Aya (human-generated), Dolly (post-edited), Native (human-generated), and Okapi (synthetic + machine translated).
- **Languages**: Arabic, English, Portuguese, Turkish, Chinese, Spanish, French, Hindi, Russian, Japanese, Korean, German, Indonesian, Italian, Vietnamese.

All main benchmarking and confusion point replacement experiments are run on the monolingual portions of LCB, using 100 prompts per language per dataset as described in Table 1.

Robustness and Generalization Experiments To assess the robustness and generalization of neuron editing, we evaluate on:

- **XNLI** (Conneau et al., 2018): Cross-lingual natural language inference in 15 languages.

- **Multilingual Sentiment Analysis**: Standard multilingual sentiment datasets (including German, Spanish, French, Japanese, and Chinese). It is a binary classification task derived from the multilingual Amazon review dataset.
- **Out-of-domain LCB evaluation**: For each language, neurons are selected from one LCB source (e.g., Aya), then tested on a different source (e.g., Okapi) to assess generalization.

H.2.3 Metrics

Language Confusion Metrics We adopt two primary metrics from LCB:

- **Line-level Pass Rate (LPR)**: Percentage of responses where every line is in the correct language.
- **Line-level Accuracy**: Proportion of lines generated in the correct language.

Language identification for these metrics is performed using the fastText classifier (Joulin et al., 2016).

Internal Model Metrics We further report:

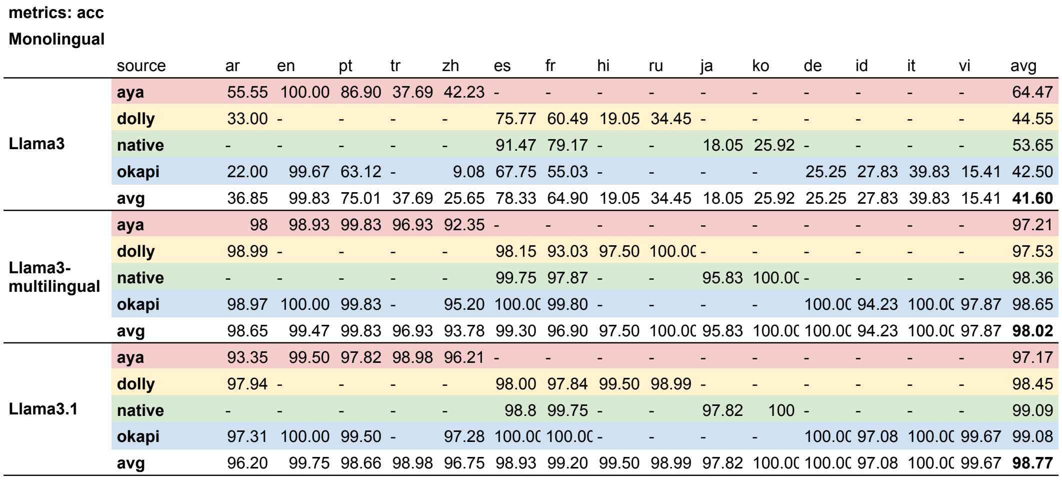
- **Target Language Token Count**: Number of target language tokens among the top-10 output logits in the final layer.
- **Target Language Token Probability**: Total probability mass assigned to target language tokens in the top-10 output logits.

Generalization and Fluency Metrics

- **XNLI and Sentiment Accuracy**: Standard classification accuracy on XNLI and multilingual sentiment analysis tasks.
- **Fluency (Perplexity)**: Perplexity of generated outputs, measured using the multilingual `facebook/xglm-564M` model (Lin et al., 2022).

H.2.4 Implementation Details

All experiments are run on NVIDIA A100 GPUs. Prompt formatting and decoding settings follow the LCB benchmark defaults. Neuron interventions are implemented at inference time via custom hooks in PyTorch, zeroing out selected neuron activations layer-wise as described in Section 5.1. For TunedLens analysis, we use the public implementation from Belrose et al. (2023).

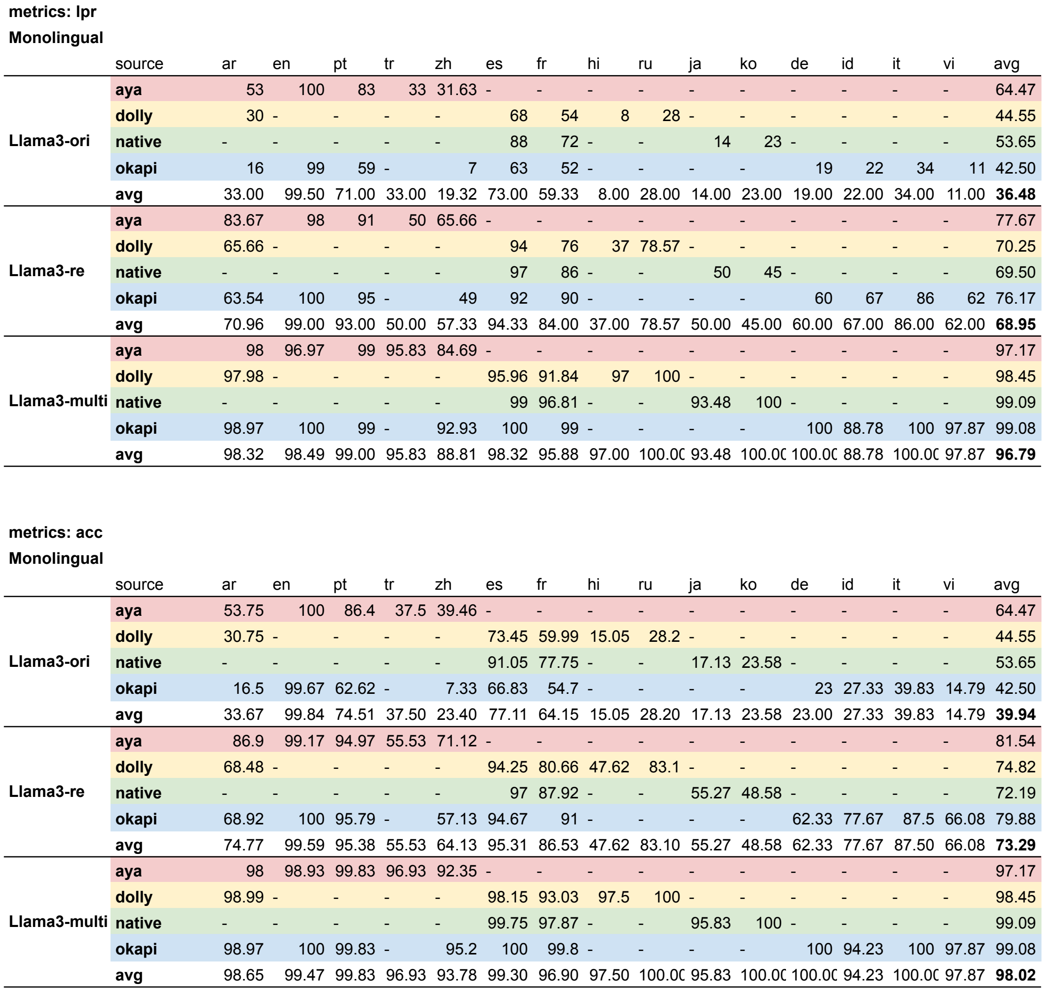


metrics: acc

Monolingual

	source	ar	en	pt	tr	zh	es	fr	hi	ru	ja	ko	de	id	it	vi	avg
Llama3	aya	55.55	100.00	86.90	37.69	42.23	-	-	-	-	-	-	-	-	-	-	64.47
	dolly	33.00	-	-	-	-	75.77	60.49	19.05	34.45	-	-	-	-	-	-	44.55
	native	-	-	-	-	-	91.47	79.17	-	-	18.05	25.92	-	-	-	-	53.65
	okapi	22.00	99.67	63.12	-	9.08	67.75	55.03	-	-	-	-	25.25	27.83	39.83	15.41	42.50
	avg	36.85	99.83	75.01	37.69	25.65	78.33	64.90	19.05	34.45	18.05	25.92	25.25	27.83	39.83	15.41	**41.60**
Llama3-multilingual	aya	98	98.93	99.83	96.93	92.35	-	-	-	-	-	-	-	-	-	-	97.21
	dolly	98.99	-	-	-	-	98.15	93.03	97.50	100.0(	-	-	-	-	-	-	97.53
	native	-	-	-	-	-	99.75	97.87	-	-	95.83	100.0(	-	-	-	-	98.36
	okapi	98.97	100.00	99.83	-	95.20	100.0(	99.80	-	-	-	-	100.0(	94.23	100.0(	97.87	98.65
	avg	98.65	99.47	99.83	96.93	93.78	99.30	96.90	97.50	100.0(	95.83	100.0(	100.0(	94.23	100.0(	97.87	**98.02**
Llama3.1	aya	93.35	99.50	97.82	98.98	96.21	-	-	-	-	-	-	-	-	-	-	97.17
	dolly	97.94	-	-	-	-	98.00	97.84	99.50	98.99	-	-	-	-	-	-	98.45
	native	-	-	-	-	-	98.8	99.75	-	-	97.82	100	-	-	-	-	99.09
	okapi	97.31	100.00	99.50	-	97.28	100.0(	100.0(	-	-	-	-	100.0(	97.08	100.0(	99.67	99.08
	avg	96.20	99.75	98.66	98.98	96.75	98.93	99.20	99.50	98.99	97.82	100.0(	100.0(	97.08	100.0(	99.67	**98.77**

Table 19: Full benchmarking results on LCB.



metrics: lpr

Monolingual

	source	ar	en	pt	tr	zh	es	fr	hi	ru	ja	ko	de	id	it	vi	avg
Llama3-ori	aya	53	100	83	33	31.63	-	-	-	-	-	-	-	-	-	-	64.47
	dolly	30	-	-	-	-	68	54	8	28	-	-	-	-	-	-	44.55
	native	-	-	-	-	-	88	72	-	-	14	23	-	-	-	-	53.65
	okapi	16	99	59	-	7	63	52	-	-	-	-	19	22	34	11	42.50
	avg	33.00	99.50	71.00	33.00	19.32	73.00	59.33	8.00	28.00	14.00	23.00	19.00	22.00	34.00	11.00	**36.48**
Llama3-re	aya	83.67	98	91	50	65.66	-	-	-	-	-	-	-	-	-	-	77.67
	dolly	65.66	-	-	-	-	94	76	37	78.57	-	-	-	-	-	-	70.25
	native	-	-	-	-	-	97	86	-	-	50	45	-	-	-	-	69.50
	okapi	63.54	100	95	-	49	92	90	-	-	-	-	60	67	86	62	76.17
	avg	70.96	99.00	93.00	50.00	57.33	94.33	84.00	37.00	78.57	50.00	45.00	60.00	67.00	86.00	62.00	**68.95**
Llama3-multi	aya	98	96.97	99	95.83	84.69	-	-	-	-	-	-	-	-	-	-	97.17
	dolly	97.98	-	-	-	-	95.96	91.84	97	100	-	-	-	-	-	-	98.45
	native	-	-	-	-	-	99	96.81	-	-	93.48	100	-	-	-	-	99.09
	okapi	98.97	100	99	-	92.93	100	99	-	-	-	-	100	88.78	100	97.87	99.08
	avg	98.32	98.49	99.00	95.83	88.81	98.32	95.88	97.00	100.0(	93.48	100.0(	100.0(	88.78	100.0(	97.87	**96.79**

metrics: acc

Monolingual

	source	ar	en	pt	tr	zh	es	fr	hi	ru	ja	ko	de	id	it	vi	avg
Llama3-ori	aya	53.75	100	86.4	37.5	39.46	-	-	-	-	-	-	-	-	-	-	64.47
	dolly	30.75	-	-	-	-	73.45	59.99	15.05	28.2	-	-	-	-	-	-	44.55
	native	-	-	-	-	-	91.05	77.75	-	-	17.13	23.58	-	-	-	-	53.65
	okapi	16.5	99.67	62.62	-	7.33	66.83	54.7	-	-	-	-	23	27.33	39.83	14.79	42.50
	avg	33.67	99.84	74.51	37.50	23.40	77.11	64.15	15.05	28.20	17.13	23.58	23.00	27.33	39.83	14.79	**39.94**
Llama3-re	aya	86.9	99.17	94.97	55.53	71.12	-	-	-	-	-	-	-	-	-	-	81.54
	dolly	68.48	-	-	-	-	94.25	80.66	47.62	83.1	-	-	-	-	-	-	74.82
	native	-	-	-	-	-	97	87.92	-	-	55.27	48.58	-	-	-	-	72.19
	okapi	68.92	100	95.79	-	57.13	94.67	91	-	-	-	-	62.33	77.67	87.5	66.08	79.88
	avg	74.77	99.59	95.38	55.53	64.13	95.31	86.53	47.62	83.10	55.27	48.58	62.33	77.67	87.50	66.08	**73.29**
Llama3-multi	aya	98	98.93	99.83	96.93	92.35	-	-	-	-	-	-	-	-	-	-	97.17
	dolly	98.99	-	-	-	-	98.15	93.03	97.5	100	-	-	-	-	-	-	98.45
	native	-	-	-	-	-	99.75	97.87	-	-	95.83	100	-	-	-	-	99.09
	okapi	98.97	100	99.83	-	95.2	100	99.8	-	-	-	-	100	94.23	100	97.87	99.08
	avg	98.65	99.47	99.83	96.93	93.78	99.30	96.90	97.50	100.0(	95.83	100.0(	100.0(	94.23	100.0(	97.87	**98.02**

Table 20: Full results of CP replacement experiments

	num_ori	prob_ori	num_edit	prob_edit	num_diff	prob_diff	fluency_ori	fluency_cna	diff
ar	2.83	25.8	5.37	30.3	2.55	4.5	30.1	24.7	-5.4
pt	2.86	49.5	3.41	56.0	0.56	6.5	25.7	23.3	-2.3
tr	2.05	29.5	2.42	23.5	0.37	-6.0	21.2	18.8	-2.5
zh	1.33	8.6	5.10	37.3	3.78	28.7	33.1	26.0	-7.0
es	1.67	26.5	3.28	50.3	1.61	23.8	25.4	23.2	-2.2
fr	2.48	43.0	2.91	49.2	0.43	6.2	21.2	21.1	-0.1
hi	1.25	12.0	1.64	13.7	0.39	1.8	28.5	22.9	-5.6
ru	1.09	18.0	3.21	31.0	2.12	13.0	23.7	19.5	-4.2
de	2.73	23.7	4.45	37.1	1.72	13.4	23.8	18.5	-5.3
it	1.33	8.4	2.50	39.3	1.17	31.0	25.7	20.2	-5.5
avg	1.96	24.5	3.43	36.8	1.47	12.3	25.8	21.8	-4.0

Table 21: Full results of robustness experiments. Perplexity is calculated to measure fluency.

xnli		
language	acc_ori	acc_edit
ar	0.42	0.37
de	0.54	0.54
es	0.46	0.5
fr	0.49	0.5
hi	0.47	0.48
ru	0.37	0.3
tr	0.46	0.52
vi	0.46	0.37
zh	0.51	0.46
avg	0.464	0.449

sentiment analysis		
language	acc_ori	acc_edit
de	0.98	0.98
es	0.98	0.98
fr	0.98	0.97
ja	0.99	0.99
zh	0.99	0.99
avg	0.984	0.982

Table 22: Full results of generalization experiments.

List of Figures

List of Tables

Bibliography

(2020). *Neural machine translation with byte-level subwords*, volume 34.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. (2020). Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Agić, Ž., Johannsen, A., Plank, B., Alonso, H. M., Schluter, N., and Søgaard, A. (2016). Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Ahuja, K., Diddee, H., Hada, R., Ochieng, M., Ramesh, K., Jain, P., Nambi, A., Ganu, T., Segal, S., Ahmed, M., Bali, K., and Sitaram, S. (2023). MEGA: Multilingual evaluation of generative AI. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

Ahuja, K., Sitaram, S., Dandapat, S., and Choudhury, M. (2022). On the calibration of massively multilingual language models. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4310–4323, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

AI, M. (2024). Introducing Meta Llama 3: The most capable openly available LLM to date.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. (2023). GQA: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., et al. (2023). Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Artetxe, M., Ruder, S., and Yogatama, D. (2020). On the cross-lingual transferability of monolingual representations. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Artetxe, M. and Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Asai, A., Kudugunta, S., Yu, X., Blevins, T., Gonen, H., Reid, M., Tsvetkov, Y., Ruder, S., and Hajishirzi, H. (2024). BUFFET: Benchmarking large language models for few-shot cross-lingual transfer. In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1771–1800, Mexico City, Mexico. Association for Computational Linguistics.

Assael, Y., Sommerschield, T., Shillingford, B., Bordbar, M., Pavlopoulos, J., Chatzipanagiotou, M., Androutsopoulos, I., Prag, J., and de Freitas, N. (2022). Restoring and attributing ancient texts using deep neural networks. *Nature*, 603(7900):280–283.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. (2023). Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bang, N., Lee, J., and Koo, M.-W. (2023a). Task-optimized adapters for an end-to-end task-oriented dialogue system. *arXiv preprint arXiv:2305.02468*.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q. V., Xu, Y., and Fung, P. (2023b). A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In Park, J. C., Arase, Y., Hu, B., Lu, W., Wijaya, D., Purwarianti, A., and Krisnadhi, A. A., editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Bau, D., Zhu, J.-Y., Strobelt, H., Lapedriza, A., Zhou, B., and Torralba, A. (2020). Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078.

Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Belinkov, Y., Gehrmann, S., and Pavlick, E. (2020). Interpretability and analysis in neural NLP. In Savary, A. and Zhang, Y., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 1–5, Online. Association for Computational Linguistics.

Belinkov, Y. and Glass, J. (2019). Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Belrose, N., Furman, Z., Smith, L., Halawi, D., Ostrovsky, I., McKinney, L., Biderman, S., and Steinhardt, J. (2023). Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Ben Zaken, E., Goldberg, Y., and Ravfogel, S. (2022). BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Bender, E. M. and Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Beniwal, H., D, K., and Singh, M. (2024). Cross-lingual editing in multilingual language models. In Graham, Y. and Purver, M., editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2078–2128, St. Julian's, Malta. Association for Computational Linguistics.

Beyer, A., Loáiciga, S., and Schlangen, D. (2021). Is incoherence surprising? targeted evaluation of coherence prediction from language models. *arXiv preprint arXiv:2105.03495*.

Blanco-Elorrieta, E. and Pylkkänen, L. (2017). Bilingual language switching in the laboratory versus in the wild: The spatiotemporal dynamics of adaptive language control. *Journal of Neuroscience*, 37(37):9022–9036.

Blevins, T., Gonen, H., and Zettlemoyer, L. (2023). Prompting language models for linguistic structure. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663, Toronto, Canada. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Booth, H., Breitbarth, A., Ecay, A., and Farasyn, M. (2020). A Penn-style treebank of Middle Low German. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 766–775, Marseille, France. European Language Resources Association.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, volume 168, pages 24–41.

Brennan, J. R. (2022). *Language and the brain: A slim guide to neurolinguistics*. Oxford University Press.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Bullock, B. E. and Toribio, A. J. E. (2009). *The Cambridge handbook of linguistic code-switching.* Cambridge university press.

Cao, B., Lin, H., Han, X., Sun, L., Yan, L., Liao, M., Xue, T., and Xu, J. (2021). Knowledgeable or educated guess? revisiting language models as knowledge bases. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *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 1860–1874, Online. Association for Computational Linguistics.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder for English. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174.

Chan, B., Schweter, S., and Möller, T. (2020). German's next language model. In Scott, D., Bel, N., and Zong, C., editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chang, T., Tu, Z., and Bergen, B. (2022). The geometry of multilingual language model representations. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 119–136, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chatterjee, A., Narahari, K. N., Joshi, M., and Agrawal, P. (2019). SemEval-2019 task 3: EmoContext contextual emotion detection in text. In May, J., Shutova, E., Herbelot, A., Zhu, X., Apidianaki, M., and Mohammad, S. M., editors, *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Chen, P., Ji, S., Bogoychev, N., Haddow, B., and Heafield, K. (2023). Monolingual or multilingual instruction tuning: Which makes a better Alpaca. *arXiv preprint arXiv:2309.08958*.

Chen, X. and Cardie, C. (2018). Multinomial adversarial networks for multi-domain text classification. *arXiv preprint arXiv:1802.05694*.

Chi, Z., Dong, L., Wei, F., Yang, N., Singhal, S., Wang, W., Song, X., Mao, X.-L., Huang, H., and Zhou, M. (2021). InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Chiarcos, C., Kosmehl, B., Fäth, C., and Sukhareva, M. (2018). Analyzing Middle High German syntax with RDF and SPARQL. In Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Chidambaram, M., Yang, Y., Cer, D., Yuan, S., Sung, Y.-H., Strope, B., and Kurzweil, R. (2018). Learning cross-lingual sentence representations via a multi-task dual-encoder model. *arXiv preprint arXiv:1810.12836*.

Child, R., Gray, S., Radford, A., and Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Chirkova, N. and Nikoulina, V. (2024). Key ingredients for effective zero-shot cross-lingual knowledge transfer in generative tasks. In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7222–7238, Mexico City, Mexico. Association for Computational Linguistics.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2023). Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Chowdhury, J. R., Zhuang, Y., and Wang, S. (2022). Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *AAAI*.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., and Palomaki, J. (2020). TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Cohen, R., Biran, E., Yoran, O., Globerson, A., and Geva, M. (2024). Evaluating the Ripple Effects of Knowledge Editing in Language Models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain. Association for Computational Linguistics.

Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. (2023). Towards automated circuit discovery for mechanistic interpretability. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H., and Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Cross, J. and Huang, L. (2016). Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas. Association for Computational Linguistics.

Cui, L., Wu, Y., Liu, J., Yang, S., and Zhang, Y. (2021). Template-based named entity recognition using BART. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Dagan, I., Glickman, O., and Magnini, B. (2005). The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. (2022). Knowledge neurons in pretrained transformers. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Dar, G., Geva, M., Gupta, A., and Berant, J. (2023). Analyzing transformers in embedding space. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16124–16170, Toronto, Canada. Association for Computational Linguistics.

De Cao, N., Aziz, W., and Titov, I. (2021). Editing factual knowledge in language models. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

De Saussure, F. (1989). *Cours de linguistique générale*, volume 1. Otto Harrassowitz Verlag.

Demske, U. (2019). Referenzkorpus frühneuhochdeutsch: Baumbank. *UP. Universität Potsdam: Institut für Germanistik (https://hdl. handle. net/11022/0000-0007-EAF7-B).*

Deng, Y., Zhang, W., Pan, S. J., and Bing, L. (2023). Multilingual jailbreak challenges in large language models. *arXiv preprint arXiv:2310.06474*.

Deshpande, A., Talukdar, P., and Narasimhan, K. (2022). When is BERT multilingual? Isolating crucial ingredients for cross-lingual transfer. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3610–3623, Seattle, United States. Association for Computational Linguistics.

Devine, P. (2024). Tagengo: A multilingual chat dataset. In Sälevä, J. and Owodunni, A., editors, *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 106–113, Miami, Florida, USA. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *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.

Dhingra, B., Cole, J. R., Eisenschlos, J. M., Gillick, D., Eisenstein, J., and Cohen, W. W. (2022). Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., et al. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Dipper, S., Donhauser, K., Klein, T., Linde, S., Müller, S., and Wegera, K.-P. (2013). HiTS: ein Tagset für historische Sprachstufen des Deutschen. *Journal for Language Technology and Computational Linguistics*, 28(1):85–137.

Doğruöz, A. S., Sitaram, S., Bullock, B. E., and Toribio, A. J. (2021). A survey of code-switching: Linguistic and social perspectives for language technologies. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *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 1654–1666, Online. Association for Computational Linguistics.

Dolan, B. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., Sun, X., Li, L., and Sui, Z. (2024). A survey on in-context learning. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Dufter, P. and Schütze, H. (2020). Identifying elements essential for BERT's multilinguality. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Etxaniz, J., Azkune, G., Soroa, A., Lopez de Lacalle, O., and Artetxe, M. (2024). Do multilingual language models think better in English? In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 550–564, Mexico City, Mexico. Association for Computational Linguistics.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., et al. (2021). Beyond English-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. (2022). Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Fey, M. and Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Field, J. (2004). *Psycholinguistics: The key concepts*. Psychology Press.

Fierro, C., Foroutan, N., Elliott, D., and Søgaard, A. (2025). How do multilingual language models remember facts? *arXiv preprint arXiv:2410.14387*.

Firestone, C. (2020). Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences*, 117(43):26562–26571.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Friederici, A. D. (2011). The brain basis of language processing: from structure to function. *Physiological reviews*, 91(4):1357–1392.

Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., and Levy, R. (2019). Neural language models as psycholinguistic subjects: Representations of syntactic state. *arXiv preprint arXiv:1903.03260*.

Gaddy, D., Stern, M., and Klein, D. (2018). What's going on in neural constituency parsers? an analysis. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 999–1010, New Orleans, Louisiana. Association for Computational Linguistics.

Gage, P. (1994). A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Gao, T., Fisch, A., and Chen, D. (2021). Making pre-trained language models better few-shot learners. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *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 3816–3830, Online. Association for Computational Linguistics.

Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). The WebNLG challenge: Generating text from RDF data. In Alonso, J. M., Bugarín, A., and Reiter, E., editors, *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Gardner-Chloros, P. (2009). *Code-switching*. Cambridge university press.

Gauthier, J., Hu, J., Wilcox, E., Qian, P., and Levy, R. (2020). Syntaxgym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76.

Geva, M., Bastings, J., Filippova, K., and Globerson, A. (2023). Dissecting recall of factual associations in auto-regressive language models. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Geva, M., Caciularu, A., Wang, K., and Goldberg, Y. (2022). Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Geva, M., Schuster, R., Berant, J., and Levy, O. (2021). Transformer feed-forward layers are key-value memories. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gouws, S. and Søgaard, A. (2015). Simple task-specific bilingual word embeddings. In Mihalcea, R., Chai, J., and Sarkar, A., editors, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390, Denver, Colorado. Association for Computational Linguistics.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gupta, A., Boleda, G., Baroni, M., and Padó, S. (2015). Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21.

Guzzardo Tamargo, R. E., Valdés Kroff, J. R., and Dussias, P. E. (2016). Examining the relationship between comprehension and production processes in code-switched language. *Journal of Memory and Language*, 89:138–161. Speaking and Listening: Relationships Between Language Production and Comprehension.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Harald, H., Forkel, R., Haspelmath, M., and Bank, S. (2015). glottolog-data: Glottolog database 2.6.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Hartmann, M., de Lhoneux, M., Hershcovich, D., Kementchedjhieva, Y., Nielsen, L., Qiu, C., and Søgaard, A. (2021). A multilingual benchmark for probing negation-awareness with minimal pairs. In Bisazza, A. and Abend, O., editors, *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 244–257, Online. Association for Computational Linguistics.

Hathaliya, J. J. and Tanwar, S. (2020). An exhaustive survey on security and privacy issues in healthcare 4.0. *Computer Communications*, 153:311–335.

Haviv, A., Berant, J., and Globerson, A. (2021). BERTese: Learning to speak to BERT. *CoRR*, abs/2103.05327.

He, L., Chen, P., Nie, E., Li, Y., and Brennan, J. R. (2024a). Decoding probing: Revealing internal linguistic structures in neural language models using minimal pairs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4488–4497.

He, L., Nie, E., Schmid, H., Schütze, H., Mesgarani, N., and Brennan, J. (2024b). Large language models as neurolinguistic subjects: Identifying internal representations for form and meaning. *arXiv preprint arXiv:2411.07533*.

He, Z., He, Y., Wu, Q., and Chen, J. (2020). Fg2seq: Effectively encoding knowledge for end-to-end task-oriented dialog. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8029–8033. IEEE.

Hermjakob, U. (2001). Parsing and question classification for question answering. In *Proceedings of the ACL 2001 Workshop on Open-Domain Question Answering*.

Hewitt, J. and Liang, P. (2019). Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*.

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

Hirschmann, H. and Linde, S. (2023). Deutsche Diachrone Baumbank (Version 1.0). Humboldt-Universität zu Berlin.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hofstadter, D. R. (1995). *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought.* Basic books.

Holtzman, A., West, P., Shwartz, V., Choi, Y., and Zettlemoyer, L. (2021). Surface form competition: Why the highest probability answer isn't always right. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hotelling, H. (1992). Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Hovy, E., Gerber, L., Hermjakob, U., Lin, C.-Y., and Ravichandran, D. (2001). Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hu, J., Gauthier, J., Qian, P., Wilcox, E., and Levy, R. P. (2020a). A systematic assessment of syntactic generalization in neural language models. *arXiv preprint arXiv:2005.03692*.

Hu, J. and Levy, R. (2023). Prompting is not a substitute for probability measurements in large language models. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. (2020b). XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Huang, L., Ma, S., Zhang, D., Wei, F., and Wang, H. (2022). Zero-shot cross-lingual transfer of prompt-based tuning with a unified multilingual prompt. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11488–11497, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jiang, J. (2012). Information extraction from text. *Mining text data*, pages 11–41.

Jiang, M. and Diesner, J. (2019). A constituency parsing tree based method for relation extraction from abstracts of scholarly publications. In Ustalov, D., Somasundaran, S., Jansen, P., Glavaš, G., Riedl, M., Surdeanu, M., and Vazirgiannis, M., editors, *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 186–191, Hong Kong. Association for Computational Linguistics.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jones, H. and Jones, M. H. (2019). *The Oxford Guide to Middle High German*. Oxford University Press.

Joshi, P., Santy, S., Budhiraja, A., Bali, K., and Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the NLP world. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). FastText.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In Lapata, M., Blunsom, P., and Koller, A., editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Jundi, I. and Lapesa, G. (2022). How to translate your samples and choose your shots? Analyzing translate-train & few-shot cross-lingual transfer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 129–150.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 1st edition.

Kaing, H., Ding, C., Utiyama, M., Sumita, E., Sudoh, K., and Nakamura, S. (2021). Constituency parsing by cross-lingual delexicalization. *IEEE Access*, 9:141571–141578.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Kassner, N., Dufter, P., and Schütze, H. (2021). Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258.

Kauf, C., Ivanova, A. A., Rambelli, G., Chersoni, E., She, J. S., Chowdhury, Z., Fedorenko, E., and Lenci, A. (2023). Event knowledge in large language models: The gap between the impossible and the unlikely. *Cognitive Science*, 47(11):e13386.

Kemmerer, D. (2022). *Cognitive neuroscience of language*. Routledge.

Keung, P., Lu, Y., Szarvas, G., and Smith, N. A. (2020). The multilingual Amazon reviews corpus. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Kew, T., Schottmann, F., and Sennrich, R. (2024). Turning English-centric LLMs into polyglots: How much multilinguality is needed? In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13097–13124, Miami, Florida, USA. Association for Computational Linguistics.

Khanuja, S., Dandapat, S., Srinivasan, A., Sitaram, S., and Choudhury, M. (2020). GLUECoS: An evaluation benchmark for code-switched NLP. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Khodja, H. A., Bechet, F., Brabant, Q., Nasr, A., and Lecorvé, G. (2024). WikiFactDiff: A large, realistic, and temporally adaptable dataset for atomic factual knowledge update in causal language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17614–17624.

Kim, D., Seo, M., Park, K., Shin, I., Woo, S., Kweon, I. S., and Choi, D.-G. (2023). Bidirectional domain mixup for domain adaptive semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1114–1123.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. *Advances in neural information processing systems*, 28.

Kitaev, N., Cao, S., and Klein, D. (2019). Multilingual constituency parsing with self-attention and pre-training. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Kitaev, N. and Klein, D. (2018). Constituency parsing with a self-attentive encoder. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Kitaev, N. and Klein, D. (2020). Tetra-tagging: Word-synchronous parsing with linear-time inference. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6255–6261, Online. Association for Computational Linguistics.

Klein, T., Wegera, K.-P., Dipper, S., and Wich-Reif, C. (2016). Reference Corpus of Middle High German (1050–1350) (Version 1.0). Rheinische Friedrich-Wilhelms-Universität Bonn, Ruhr-Universität Bochum.

Ko, D., Lee, J., Kang, W.-Y., Roh, B., and Kim, H. (2023). Large language models are temporal and causal reasoners for video question answering. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4300–4316, Singapore. Association for Computational Linguistics.

Köhn, A. (2015). What's in an embedding? Analyzing word embeddings through multilingual evaluation. In Màrquez, L., Callison-Burch, C., and Su, J., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal. Association for Computational Linguistics.

Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., and Hajishirzi, H. (2019). Text Generation from Knowledge Graphs with Graph Transformers. In Burstein, J., Doran, C., and Solorio, T., editors, *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 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Lai, G. and O'Brien, B. A. (2020). Examining language switching and cognitive control through the adaptive control hypothesis. *Frontiers in Psychology*, 11:1171.

Lai, V., Ngo, N., Pouran Ben Veyseh, A., Man, H., Dernoncourt, F., Bui, T., and Nguyen, T. (2023a). ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the*

Association for Computational Linguistics: EMNLP 2023, pages 13171–13189, Singapore. Association for Computational Linguistics.

Lai, V., Nguyen, C., Ngo, N., Nguyen, T., Dernoncourt, F., Rossi, R., and Nguyen, T. (2023b). Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In Feng, Y. and Lefever, E., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 318–327, Singapore. Association for Computational Linguistics.

Lampinen, A. (2024). Can language models handle recursively nested grammatical structures? A case study on comparing models and humans. *Computational Linguistics*, pages 1–36.

Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *The Eighth International Conference on Learning Representations*.

Lauscher, A., Ravishankar, V., Vulić, I., and Glavaš, G. (2020). From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Levy, O., Seo, M., Choi, E., and Zettlemoyer, L. (2017). Zero-shot relation extraction via reading comprehension. In Levy, R. and Specia, L., editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lewis, P., Oğuz, B., Rinott, R., Riedel, S., and Schwenk, H. (2019). MLQA: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Li, B. Z., Chen, W., Sharma, P., and Andreas, J. (2023a). LaMPP: Language models as probabilistic priors for perception and action. *arXiv e-prints*, pages arXiv–2302.

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., and Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Li, J., Tang, T., Zhao, W. X., Wei, Z., Yuan, N. J., and Wen, J.-R. (2021). Few-shot knowledge graph-to-text generation with pretrained language models. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1558–1568, Online. Association for Computational Linguistics.

Li, S., Jiang, W., Si, P., Yang, C., Yao, Q., Zhang, J., Zhou, J., and Yang, Y. (2023b). Enhancing dialogue generation with conversational concept flows. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1484–1495.

Li, T. and Murray, K. (2023). Why does zero-shot cross-lingual generation fail? An explanation and a solution. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12461–12476, Toronto, Canada. Association for Computational Linguistics.

Li, X., Li, M., Wang, Y., Ren, C.-X., and Guo, X. (2023c). Adaptive texture filtering for single-domain generalized segmentation. *arXiv preprint arXiv:2303.02943*.

Li, X. and Roth, D. (2002). Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Li, X. L. and Liang, P. (2021). Prefix-Tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

Li, Y., Yin, C., and Zhong, S.-h. (2020). Sentence constituent-aware aspect-category sentiment analysis with graph attention networks. In *Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, October 14–18, 2020, Proceedings, Part I 9*, pages 815–827. Springer.

Liang, S., Dufter, P., and Schütze, H. (2021). Locating language-specific information in contextualized embeddings. *arXiv preprint arXiv:2109.08040*.

Lin, X. V., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., Ott, M., Goyal, N., Bhosale, S., Du, J., Pasunuru, R., Shleifer, S., Koura, P. S., Chaudhary, V., O'Horo, B., Wang, J., Zettlemoyer, L., Kozareva, Z., Diab, M., Stoyanov, V., and Li, X. (2022). Few-shot learning with multilingual generative language models. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lin, Z., Madotto, A., Winata, G. I., Xu, P., Jiang, F., Hu, Y., Shi, C., and Fung, P. (2021). Bitod: A bilingual multi-domain dataset for task-oriented dialogue modeling. *arXiv preprint arXiv:2106.02787*.

Linzen, T., Dupoux, E., and Goldberg, Y. (2016). Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Littell, P., Mortensen, D. R., Lin, K., Kairis, K., Turner, C., and Levin, L. (2017). URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In Lapata, M., Blunsom, P., and Koller, A., editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. (2022a). What makes good in-context examples for GPT-3? In Agirre, E., Apidianaki, M., and Vulić, I., editors, *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019a). Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Liu, W., Tang, J., Cheng, Y., Li, W., Zheng, Y., and Liang, X. (2022b). MedDG: An entity-centric medical consultation dataset for entity-aware medical dialogue generation. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. (2022c). P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Liu, Y., Feng, S., Wang, D., and Zhang, Y. (2022d). MulZDG: Multilingual code-switching framework for zero-shot dialogue generation. In Calzolari, N., Huang, C.-R., Kim, H., Pustejovsky, J., Wanner, L., Choi, K.-S., Ryu, P.-M., Chen, H.-H., Donatelli, L., Ji, H., Kurohashi, S., Paggio, P., Xue, N., Kim, S., Hahm, Y., He, Z., Lee, T. K., Santus, E., Bond, F., and Na, S.-H., editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 648–659, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Liu, Y., Feng, S., Wang, D., Zhang, Y., and Schütze, H. (2023b). Evaluate what you can't evaluate: Unassessable generated responses quality. *arXiv preprint arXiv:2305.14658*.

Liu, Y., Feng, S., Wang, D., Zhang, Y., and Schütze, H. (2023c). PVGRU: Generating diverse and relevant dialogue responses via pseudo-variational mechanism. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3295–3310, Toronto, Canada. Association for Computational Linguistics.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Y., Schick, T., and Schütze, H. (2022e). Semantic-oriented unlabeled priming for large-scale language models. *arXiv preprint arXiv:2202.06133*.

Liu, Y., Ye, H., Weissweiler, L., Wicke, P., Pei, R., Zangenfeind, R., and Schütze, H. (2023d). A crosslingual investigation of conceptualization in 1335 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12969–13000, Toronto, Canada. Association for Computational Linguistics.

Liu, Y., Zhang, Y., Li, Q., Feng, S., Wang, D., Zhang, Y., and Schütze, H. (2024). HiFT: A hierarchical full parameter fine-tuning strategy. *arXiv preprint arXiv:2401.15207*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *International Conference on Learning Representations*.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Ma, B., Nie, E., Schmid, H., and Schütze, H. (2023a). Is prompt-based finetuning always better than vanilla finetuning? insights from cross-lingual language understanding. In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2023)*, Ingolstadt, Germany. KONVENS 2023 Organizers.

Ma, B., Nie, E., Yuan, S., Schmid, H., Färber, M., Kreuter, F., and Schuetze, H. (2024). ToPro: Token-level prompt decomposition for cross-lingual sequence labeling tasks. In Graham, Y. and Purver, M., editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2685–2702, St. Julian's, Malta. Association for Computational Linguistics.

Ma, R., Zhou, X., Gui, T., Tan, Y., Li, L., Zhang, Q., and Huang, X. (2022). Template-free prompt tuning for few-shot NER. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States. Association for Computational Linguistics.

Ma, X., Zhang, P., and Zhao, F. (2023b). Domain-specific attention with distributional signatures for multi-domain end-to-end task-oriented dialogue. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3109–3122.

Madotto, A., Wu, C.-S., and Fung, P. (2018). Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478.

Malaviya, C., Neubig, G., and Littell, P. (2017). Learning language representations for typology prediction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark.

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J. D., Chen, D., and Arora, S. (2023). Fine-tuning language models with just forward passes. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. (2022). PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Manning, C. D., Clark, K., Hewitt, J., Khandelwal, U., and Levy, O. (2020). Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Manning, C. D., Schütze, H., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Marchisio, K., Ko, W.-Y., Berard, A., Dehaze, T., and Ruder, S. (2024). Understanding and mitigating language confusion in LLMs. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6653–6677, Miami, Florida, USA. Association for Computational Linguistics.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

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

May, C., Wang, A., Bordia, S., Bowman, S. R., and Rudinger, R. (2019). On measuring social biases in sentence encoders. *ArXiv*, abs/1903.10561.

McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

McDonald, R., Petrov, S., and Hall, K. (2011). Multi-source transfer of delexicalized dependency parsers. In Barzilay, R. and Johnson, M., editors, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

McInnes, L., Healy, J., Saul, N., and Großberger, L. (2018). UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29).

Mekala, D., Wolfe, J., and Roy, S. (2023). ZEROTOP: Zero-shot task-oriented semantic parsing using large language models. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5792–5799, Singapore. Association for Computational Linguistics.

Men, T., Cao, P., Jin, Z., Chen, Y., Liu, K., and Zhao, J. (2024). Unlocking the future: Exploring look-ahead planning mechanistic interpretability in large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7713–7724, Miami, Florida, USA. Association for Computational Linguistics.

Meng, K., Bau, D., Andonian, A. J., and Belinkov, Y. (2022). Locating and editing factual associations in GPT. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Meng, K., Sharma, A. S., Andonian, A. J., Belinkov, Y., and Bau, D. (2023). Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Meng, R., Liu, Y., Joty, S. R., Xiong, C., Zhou, Y., and Yavuz, S. (2024). SFR-Embedding-Mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one? In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Misra, K., Rayz, J., and Ettinger, A. (2023). COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2920–2941.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. (2021). Fast model editing at scale. In *International Conference on Learning Representations*.

Mitchell, E., Lin, C., Bosselut, A., Manning, C. D., and Finn, C. (2022). Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Mitchell, M. and Krakauer, D. C. (2023). The debate over understanding in AI's large language models. *Proceedings of the National Academy of Sciences*, 120(13):e2215907120.

Morcos, A., Raghu, M., and Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31.

Moreno, E. M., Federmeier, K. D., and Kutas, M. (2002). Switching languages, switching palabras (words): An electrophysiological study of code switching. *Brain and language*, 80(2):188–207.

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Le Scao, T., Bari, M. S., Shen, S., Yong, Z. X., Schoelkopf, H., Tang, X., Radev, D., Aji, A. F., Almubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E., and Raffel, C. (2023). Crosslingual generalization through multitask finetuning. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Nguyen, X.-P., Zhang, W., Li, X., Aljunied, M., Tan, Q., Cheng, L., Chen, G., Deng, Y., Yang, S., Liu, C., et al. (2023). SeaLLMs–Large Language Models for Southeast Asia. *arXiv preprint arXiv:2312.00738*.

Nie, E., Liang, S., Schmid, H., and Schütze, H. (2023a). Cross-lingual retrieval augmented prompt for low-resource languages. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8320–8340, Toronto, Canada. Association for Computational Linguistics.

Nie, E., Schmid, H., and Schuetze, H. (2023b). Unleashing the multilingual encoder potential: Boosting zero-shot performance via probability calibration. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15774–15782, Singapore. Association for Computational Linguistics.

Nie, E., Yuan, S., Ma, B., Schmid, H., Färber, M., Kreuter, F., and Schütze, H. (2024). Decomposed prompting: Unveiling multilingual linguistic structure knowledge in English-centric large language models. *arXiv preprint arXiv:2402.18397*.

Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., and Zeman, D. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Nostalgebraist (2020). Interpreting GPT: The logit lens.

OpenAI (2023). GPT-4 technical report.

Östling, R. and Kurfalı, M. (2023). Language embeddings sometimes contain typological generalizations. *Computational Linguistics*, 49(4):1003–1051.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Pal, K., Sun, J., Yuan, A., Wallace, B., and Bau, D. (2023). Future Lens: Anticipating subsequent tokens from a single hidden state. In Jiang, J., Reitter, D., and Deng, S., editors, *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 548–560, Singapore. Association for Computational Linguistics.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Pan, X., Zhang, B., May, J., Nothman, J., Knight, K., and Ji, H. (2017). Cross-lingual name tagging and linking for 282 languages. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Parravicini, A. and Pievani, T. (2018). Continuity and discontinuity in human language evolution: putting an old-fashioned debate in its historical perspective. *Topoi*, 37(2):279–287.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Patel, A., Bhattamishra, S., Reddy, S., and Bahdanau, D. (2023). MAGNIFICo: Evaluating the in-context learning ability of large language models to generalize to novel interpretations. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2167–2189, Singapore. Association for Computational Linguistics.

Peng, B., Li, C., He, P., Galley, M., and Gao, J. (2023). Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*.

Peng, S., Huang, X., Lin, Z., Ji, F., Chen, H., and Zhang, Y. (2019). Teacher-student framework enhanced multi-domain dialogue generation. *arXiv preprint arXiv:1908.07137*.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *NAACL*.

Pfeiffer, J., Piccinno, F., Nicosia, M., Wang, X., Reid, M., and Ruder, S. (2023). mmT5: Modular multilingual pre-training solves source language hallucinations. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1978–2008, Singapore. Association for Computational Linguistics.

Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., and Gurevych, I. (2020a). AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. (2020b). MAD-X: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual BERT? In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Poth, C., Sterz, H., Paul, I., Purkayastha, S., Engländer, L., Imhof, T., Vulić, I., Ruder, S., Gurevych, I., and Pfeiffer, J. (2023). Adapters: A unified library for parameter-efficient and modular transfer learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160, Singapore. Association for Computational Linguistics.

Qi, J., Fernández, R., and Bisazza, A. (2023). Cross-lingual consistency of factual knowledge in multilingual language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10650–10666.

Qi, K., Wan, H., Du, J., and Chen, H. (2022). Enhancing cross-lingual natural language inference by prompt-learning from cross-lingual templates. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1910–1923, Dublin, Ireland. Association for Computational Linguistics.

Qin, G. and Eisner, J. (2021). Learning how to ask: Querying LMs with mixtures of soft prompts. *arXiv*.

Qin, L., Liu, Y., Che, W., Wen, H., Li, Y., and Liu, T. (2019). Entity-consistent end-to-end task-oriented dialogue system with kb retriever. 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 133–142.

Qin, L., Xu, X., Che, W., Zhang, Y., and Liu, T. (2020). Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6344–6354.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Rai, D., Zhou, Y., Feng, S., Saparov, A., and Yao, Z. (2024). A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*.

Rama, T., Beinborn, L., and Eger, S. (2020). Probing multilingual BERT for genetic and typological signals. In *International Conference on Computational Linguistics*.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ren, F., Hou, Y., Li, Y., Pan, L., Zhang, Y., Liang, X., Liu, Y., Guo, Y., Zhao, R., Ming, R., et al. (2018). TechKG: A large-scale chinese technology-oriented knowledge graph. *arXiv preprint arXiv:1812.06722*.

Ren, F., Ning, A., Qi, M., and Lei, H. (2023). TechGPT: Technology-oriented generative pretrained transformer. *GitHub repository*.

Ribeiro, L. F. R., Zhang, Y., and Gurevych, I. (2021). Structural adapters in pretrained language models for AMR-to-Text generation. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4269–4282, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E. M., Boureau, Y.-L., et al. (2021). Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325.

Ruder, S. (2019). *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway.

Ruder, S., Vulić, I., and Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Saha, T., Ganguly, D., Saha, S., and Mitra, P. (2023). Workshop on large language models' interpretability and trustworthiness (llmit). In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 5290–5293.

Sakai, I. (1961). Syntax in universal translation. In *Proceedings of the International Conference on Machine Translation and Applied Language Analysis*, National Physical Laboratory, Teddington, UK.

Salmons, J. (2018). *A history of German: What the past reveals about today's language*. Oxford University Press.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Saphra, N. and Wiegreffe, S. (2024). Mechanistic? In Belinkov, Y., Kim, N., Jumelet, J., Mohebbi, H., Mueller, A., and Chen, H., editors, *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 480–498, Miami, Florida, US. Association for Computational Linguistics.

Sapp, C., Dakota, D., and Evans, E. (2023). Parsing early New High German: Benefits and limitations of cross-dialectal training. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 54–66, Washington, D.C. Association for Computational Linguistics.

Schick, T. and Schütze, H. (2021a). Exploiting cloze-questions for few-shot text classification and natural language inference. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Schick, T. and Schütze, H. (2021b). Few-shot text generation with natural language instructions. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Schick, T. and Schütze, H. (2021c). It's not just size that matters: Small language models are also few-shot learners. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Schiller, A., Teufel, S., and Thielen, C. (1995). Guidelines für das Tagging deutscher Textcorpora mit STTS. *Universität Stuttgart, Universität Tübingen, Germany*.

Schmid, H. (2019). Deep learning-based morphological taggers and lemmatizers for annotating historical texts. In *Proceedings of the 3rd international conference on digital access to textual cultural heritage*, pages 133–137.

Schuster, M. and Nakajima, K. (2012). Japanese and Korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

Sedgwick, P. (2012). Pearson's correlation coefficient. *BMJ*, 345.

Sejnowski, T. J. (2023). Large language models and the reverse Turing test. *Neural computation*, 35(3):309–342.

Sekine, S. and Collins, M. (1997). Evalb bracket scoring program. *URL: http://www. cs. nyu. edu/cs/projects/proteus/evalb*.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In Erk, K. and Smith, N. A., editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In Erk, K. and Smith, N. A., editors, *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.

Sennrich, R., Vamvas, J., and Mohammadshahi, A. (2024). Mitigating hallucinations and off-target machine translation with source-contrastive and language-contrastive decoding. In Graham, Y. and Purver, M., editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 21–33, St. Julian's, Malta. Association for Computational Linguistics.

Serban, I., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Shaham, U., Herzig, J., Aharoni, R., Szpektor, I., Tsarfaty, R., and Eyal, M. (2024). Multilingual instruction tuning with just a pinch of multilinguality. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2304–2317, Bangkok, Thailand. Association for Computational Linguistics.

Shao, Y., Geng, Z., Liu, Y., Dai, J., Yan, H., Yang, F., Zhe, L., Bao, H., and Qiu, X. (2021). CPT: A pre-trained unbalanced transformer for both Chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

Shapiro, N., Paullada, A., and Steinert-Threlkeld, S. (2021). A multilabel approach to morphosyntactic probing. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4486–4524, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sharkey, L., Chughtai, B., Batson, J., Lindsey, J., Wu, J., Bushnaq, L., Goldowsky-Dill, N., Heimersheim, S., Ortega, A., Bloom, J., Biderman, S., Garriga-Alonso, A., Conmy, A., Nanda, N., Rumbelow, J., Wattenberg, M., Schoots, N., Miller, J., Michaud, E. J., Casper, S., Tegmark, M., Saunders, W., Bau, D., Todd, E., Geiger, A., Geva, M., Hoogland, J., Murfet, D., and McGrath, T. (2025). Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*.

Shi, X., Padhi, I., and Knight, K. (2016). Does string-based neural MT learn source syntax? In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1526–1534.

Shi, Z. and Lipani, A. (2023). Don't stop pretraining? Make prompt-based fine-tuning powerful learner. *arXiv*.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. (2020). AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Shliazhko, O., Fenogenova, A., Tikhonova, M., Mikhailov, V., Kozlova, A., and Shavrina, T. (2022). mGPT: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

Singh, A. K. (2008). Natural language processing for less privileged languages: Where do we come from? Where are we going? In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

Singh, J., McCann, B., Socher, R., and Xiong, C. (2019). BERT is not an interlingua and the bias of tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 47–55, Hong Kong, China. Association for Computational Linguistics.

Singh, S., Vargus, F., D'souza, D., Karlsson, B., Mahendiran, A., Ko, W.-Y., Shandilya, H., Patel, J., Mataciunas, D., O'Mahony, L., Zhang, M., Hettiarachchi, R., Wilson, J., Machado, M., Moura, L., Krzemiński, D., Fadaei, H., Ergun, I., Okoh, I., Alaagib, A., Mudannayake, O., Alyafeai, Z., Chien, V., Ruder, S., Guthikonda, S., Alghamdi, E., Gehrmann, S., Muennighoff, N., Bartolo, M., Kreutzer, J., Üstün, A., Fadaee, M., and Hooker, S. (2024). Aya dataset: An open-access collection for multilingual instruction tuning. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11521–11567, Bangkok, Thailand. Association for Computational Linguistics.

Sitaram, S., Choudhury, M., Patra, B., Chaudhary, V., Ahuja, K., and Bali, K. (2023). Everything you need to know about multilingual LLMs: Towards fair, performant and reliable models for languages of the world. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 21–26, Toronto, Canada. Association for Computational Linguistics.

Smith, G. (2003). A brief introduction to the TIGER Treebank, version 1. Technical report, Universität Potsdam.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., and Bethard, S., editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Sokal, R. R. (1958). A statistical method for evaluating systematic relationships. *Univ. Kansas, Sci. Bull.*, 38:1409–1438.

Solorio, T. and Liu, Y. (2008). Learning to predict code-switching points. In Lapata, M. and Ng, H. T., editors, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 973–981, Honolulu, Hawaii. Association for Computational Linguistics.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2019). MASS: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Stern, M., Andreas, J., and Klein, D. (2017). A minimal span-based neural constituency parser. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566.

Stolfo, A., Belinkov, Y., and Sachan, M. (2023). A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. Association for Computational Linguistics.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. *Advances in neural information processing systems*, 28.

Sun, X., Li, X., Li, J., Wu, F., Guo, S., Zhang, T., and Wang, G. (2023). Text classification via large language models. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Suurmeijer, L., Parafita Couto, M. C., and Gullberg, M. (2020). Structural and extralinguistic aspects of code-switching: Evidence from papiamentu-dutch auditory sentence matching. *Frontiers in Psychology*, 11:592266.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Van Durme, B., Bowman, S. R., Das, D., et al. (2019). What do you learn from context? Probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023a). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Traxler, M. and Gernsbacher, M. A. (2011). *Handbook of psycholinguistics*. Elsevier.

Treffers-Daller, J. (2009). *Code-switching and transfer: An exploration of similarities and differences*. Cambridge University Press.

Tsvetkov, Y. (2017). Opportunities and challenges in working with low-resource languages. In *Carnegie Mellon Univ., Language Technologies Institute*.

Tu, L., Xiong, C., and Zhou, Y. (2022). Prompt-tuning can be much better than fine-tuning on cross-lingual understanding with multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5478–5485, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vamvas, J. and Sennrich, R. (2021). On the limits of minimal pairs in contrastive evaluation. *arXiv preprint arXiv:2109.07465*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations*.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Waldis, A., Perlitz, Y., Choshen, L., Hou, Y., and Gurevych, I. (2024). Holmes: A benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647.

Wan, Z., Cheng, F., Mao, Z., Liu, Q., Song, H., Li, J., and Kurohashi, S. (2023). GPT-RE: In-context learning for relation extraction using large language models. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547, Singapore. Association for Computational Linguistics.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019a). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the Seventh International Conference on Learning Representations*.

Wang, J., Liang, Y., Sun, Z., Cao, Y., and Xu, J. (2023a). Cross-lingual knowledge editing in large language models. *arXiv preprint arXiv:2309.08952*.

Wang, K. R., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. (2023b). Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Wang, L., Li, L., Dai, D., Chen, D., Zhou, H., Meng, F., Zhou, J., and Sun, X. (2023c). Label words are anchors: An information flow perspective for understanding in-context learning. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore. Association for Computational Linguistics.

Wang, M., Adel, H., Lange, L., Liu, Y., Nie, E., Strötgen, J., and Schütze, H. (2025a). Lost in multilinguality: Dissecting cross-lingual factual inconsistency in transformer language models. *arXiv preprint arXiv:2504.04264*.

Wang, S., Xie, Y., Ding, B., Gao, J., and Zhang, Y. (2025b). Language adaptation of large language models: An empirical study on LLaMA2. In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S., editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7195–7208, Abu Dhabi, UAE. Association for Computational Linguistics.

Wang, S., Xu, Y., Fang, Y., Liu, Y., Sun, S., Xu, R., Zhu, C., and Zeng, M. (2022a). Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3170–3179, Dublin, Ireland. Association for Computational Linguistics.

Wang, W., Haddow, B., and Birch, A. (2024a). Retrieval-augmented multilingual knowledge editing. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 335–354, Bangkok, Thailand. Association for Computational Linguistics.

Wang, W., Tu, Z., Chen, C., Yuan, Y., Huang, J.-t., Jiao, W., and Lyu, M. R. (2023d). All languages matter: On the multilingual safety of large language models. *arXiv preprint arXiv:2310.00905*.

Wang, X., Ruder, S., and Neubig, G. (2022b). Expanding pretrained models to thousands more languages via lexicon-based adaptation. *arXiv preprint arXiv:2203.09435*.

Wang, Y., Chen, Y., Wen, W., Sheng, Y., Li, L., and Zeng, D. D. (2024b). Unveiling factual recall behaviors of large language models through knowledge neurons. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7388–7402, Miami, Florida, USA. Association for Computational Linguistics.

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Naik, A., Ashok, A., Dhanasekaran, A. S., Arunkumar, A., Stap, D., Pathak, E., Karamanolakis, G., Lai, H., Purohit, I., Mondal, I., Anderson, J., Kuznia, K., Doshi, K., Pal, K. K., Patel, M., Moradshahi, M., Parmar, M., Purohit, M., Varshney, N., Kaza, P. R., Verma, P., Puri, R. S., Karia, R., Doshi, S., Sampat, S. K., Mishra, S., Reddy A, S., Patro, S., Dixit, T., and Shen, X. (2022c). Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wang, Z., Mayhew, S., Roth, D., et al. (2019b). Cross-lingual ability of multilingual bert: An empirical study. *arXiv preprint arXiv:1912.07840*.

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., and Bowman, S. R. (2020). BLiMP: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022a). Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022b). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Wei, Z., Deng, J., Pang, L., Ding, H., Shen, H., and Cheng, X. (2024). Mlake: Multilingual knowledge editing benchmark for large language models. *arXiv preprint arXiv:2404.04990*.

Weissweiler, L., Hofmann, V., Köksal, A., and Schütze, H. (2022). The better your syntax, the better your semantics? probing pretrained language models for the English comparative correlative. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wen, H., Liu, Y., Che, W., Qin, L., and Liu, T. (2018). Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3781–3792.

Wendler, C., Veselovsky, V., Monea, G., and West, R. (2024). Do Llamas work in English? On the latent language of multilingual transformers. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, Bangkok, Thailand. Association for Computational Linguistics.

Wichmann, S., Holman, E. W., and H, C. (2022). The ASJP database. version 20.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In Walker, M., Ji, H., and Stent, A., editors, *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.

Winata, G., Aji, A. F., Yong, Z. X., and Solorio, T. (2023). The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2936–2978, Toronto, Canada. Association for Computational Linguistics.

Winata, G. I., Madotto, A., Lin, Z., Liu, R., Yosinski, J., and Fung, P. (2021). Language models are few-shot multilingual learners. In Ataman, D., Birch, A., Conneau, A., Firat, O., Ruder, S., and Sahin, G. G., editors, *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Workshop, B., Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., et al. (2022). BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Wu, C.-S., Madotto, A., Hosseini-Asl, E., Xiong, C., Socher, R., and Fung, P. (2019a). Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*.

Wu, C.-S., Socher, R., and Xiong, C. (2019b). Global-to-local memory pointer networks for task-oriented dialogue. *arXiv preprint arXiv:1901.04713*.

Wu, H., Xu, K., Song, L., Jin, L., Zhang, H., and Song, L. (2021). Domain-adaptive pretraining methods for dialogue understanding. *arXiv preprint arXiv:2105.13665*.

Wu, H., Zhang, Y., Jin, X., Xue, Y., and Wang, Z. (2019c). Shared-private LSTM for multi-domain text classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 116–128. Springer.

Wu, S., Conneau, A., Li, H., Zettlemoyer, L., and Stoyanov, V. (2019d). Emerging cross-lingual structure in pretrained language models. *arXiv preprint arXiv:1911.01464*.

Wu, S. and Dredze, M. (2019). Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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.

Wu, S. and Dredze, M. (2020). Are all languages created equal in multilingual BERT? In Gella, S., Welbl, J., Rei, M., Petroni, F., Lewis, P., Strubell, E., Seo, M., and Hajishirzi, H., editors, *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Xiang, B., Yang, C., Li, Y., Warstadt, A., and Kann, K. (2021). CLiMP: A benchmark for Chinese language model evaluation. *arXiv preprint arXiv:2101.11131*.

Xie, T., Wu, C. H., Shi, P., Zhong, R., Scholak, T., Yasunaga, M., Wu, C.-S., Zhong, M., Yin, P., Wang, S. I., et al. (2022). UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631.

Xu, J., Ren, X., Lin, J., and Sun, X. (2018). Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3940–3949.

Xu, Y., Hou, Y., Che, W., and Zhang, M. (2023). Language anisotropic cross-lingual model editing. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5554–5569, Toronto, Canada. Association for Computational Linguistics.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *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.

Yang, A., Liu, K., Liu, J., Lyu, Y., and Li, S. (2018). Adaptations of ROUGE and BLEU to better evaluate machine reading comprehension task. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 98–104.

Yang, L., Li, J., Li, S., and Shinozaki, T. (2023a). Multi-domain dialogue state tracking with disentangled domain-slot attention. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4928–4938.

Yang, S., Kim, J., Jang, J., Ye, S., Lee, H., and Seo, M. (2023b). Improving probability-based prompt selection through unified evaluation and analysis. *arXiv preprint arXiv:2305.14877*.

Yang, Y., Zhang, Y., Tar, C., and Baldridge, J. (2019a). PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019b). XLNet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Yao, Y., Wang, P., Tian, B., Cheng, S., Li, Z., Deng, S., Chen, H., and Zhang, N. (2023). Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240.

Ye, J., Tao, X., and Kong, L. (2023). Language versatilists vs. specialists: An empirical revisiting on multilingual transfer ability. *arXiv preprint arXiv:2306.06688*.

Yim, O. and Clément, R. (2021). Acculturation and attitudes toward code-switching: A bidimensional framework. *International Journal of Bilingualism*, 25(5):1369–1388.

Yin, W., Hay, J., and Roth, D. (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Yu, C. T. and Salton, G. (1976). Precision weighting—an effective automatic indexing method. *Journal of the ACM (JACM)*, 23(1):76–88.

Yu, Z. and Ananiadou, S. (2024a). Interpreting arithmetic mechanism in large language models through comparative neuron analysis. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3293–3306, Miami, Florida, USA. Association for Computational Linguistics.

Yu, Z. and Ananiadou, S. (2024b). Neuron-level knowledge attribution in large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3267–3280, Miami, Florida, USA. Association for Computational Linguistics.

Yuan, S., Nie, E., Färber, M., Schmid, H., and Schütze, H. (2024). GNNavi: Navigating the information flow in large language models by graph neural network. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand. Association for Computational Linguistics.

Yuan, W., Neubig, G., and Liu, P. (2021). Bartscore: Evaluating generated text as text generation. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Yunjie, J., Yong, D., Yan, G., Yiping, P., Qiang, N., Baochang, M., and Xiangang, L. (2023). BELLE: Be everyone's large language model engine. *GitHub repository*.

Zeman, D., Nivre, J., Abrams, M., Aepli, N., Agić, Ž., Ahrenberg, L., Aleksandravičiūtė, G., Antonsen, L., Aplonova, K., Aranzabe, M. J., Arutie, G., Asahara, M., Ateyah, L., et al. (2019). Universal Dependencies 2.5. *LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University*.

Zhang, H., Diao, S., Lin, Y., Fung, Y. R., Lian, Q., Wang, X., Chen, Y., Ji, H., and Zhang, T. (2023a). R-Tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020a). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Zhang, M., Gautam, V., Wang, M., Alabi, J., Shen, X., Klakow, D., and Mosbach, M. (2024a). The impact of demonstrations on multilingual in-context learning: A multidimensional analysis. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7342–7371, Bangkok, Thailand. Association for Computational Linguistics.

Zhang, N., Yao, Y., and Deng, S. (2024b). Knowledge editing for large language models. In Klinger, R., Okazaki, N., Calzolari, N., and Kan, M.-Y., editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024): Tutorial Summaries*, pages 33–41, Torino, Italia. ELRA and ICCL.

Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., and Qiao, Y. (2023b). LLaMA-Adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., et al. (2023c). Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022a). OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020b). BERTScore: Evaluating text generation with BERT. *ArXiv*, abs/1904.09675.

Zhang, X., Bosselut, A., Yasunaga, M., Ren, H., Liang, P., Manning, C., and Leskovec, J. (2022b). GreaseLM: Graph REASoning Enhanced language models for question answering. In *International Conference on Representation Learning (ICLR)*.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zhang, Y., Baldridge, J., and He, L. (2019a). PAWS: Paraphrase adversaries from word scrambling. In Burstein, J., Doran, C., and Solorio, T., editors, *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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019b). ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Zhang, Z., Li, J., Zhu, P., Zhao, H., and Liu, G. (2018). Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*.

Zhao, J., Zhang, Z., Gao, L., Zhang, Q., Gui, T., and Huang, X. (2024). Llama beyond English: An empirical study on language capability transfer. *arXiv preprint arXiv:2401.01055*.

Zhao, M. and Schütze, H. (2021). Discrete and soft prompting for multilingual models. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

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

Zheng, C., Li, L., Dong, Q., Fan, Y., Wu, Z., Xu, J., and Chang, B. (2023). Can we edit factual knowledge by in-context learning? In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Zhong, C., Cheng, F., Liu, Q., Jiang, J., Wan, Z., Chu, C., Murawaki, Y., and Kurohashi, S. (2024). Beyond English-centric LLMs: What language do multilingual language models think in? *arXiv preprint arXiv:2408.10811*.

Zhong, V., Xiong, C., and Socher, R. (2018). Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467.

Zhong, Z., Friedman, D., and Chen, D. (2021). Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

Zhong, Z., Wu, Z., Manning, C., Potts, C., and Chen, D. (2023). MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., et al. (2024). LIMA: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Zhou, H., Zheng, C., Huang, K., Huang, M., and Zhu, X. (2020). KdConv: A Chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation.

Zhou, M., Li, X., Jiang, Y., and Bing, L. (2023). Enhancing cross-lingual prompting with dual prompt augmentation. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11008–11020, Toronto, Canada. Association for Computational Linguistics.

Zhu, W., Lv, Y., Dong, Q., Yuan, F., Xu, J., Huang, S., Kong, L., Chen, J., and Li, L. (2023). Extrapolating large language models to non-English by aligning languages. *arXiv preprint arXiv:2308.04948*.

Zhuang, L., Wayne, L., Ya, S., and Jun, Z. (2021). A robustly optimized BERT pre-training approach with post-training. In Li, S., Sun, M., Liu, Y., Wu, H., Liu, K., Che, W., He, S., and Rao, G., editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Ziyu, Z., Qiguang, C., Longxuan, M., Mingda, L., Yi, H., Yushan, Q., Haopeng, B., Weinan, Z., and Liu, T. (2023). Through the lens of core competency: Survey on evaluation of large language models. In Zhang, J., editor, *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 2: Frontier Forum)*, pages 88–109, Harbin, China. Chinese Information Processing Society of China.