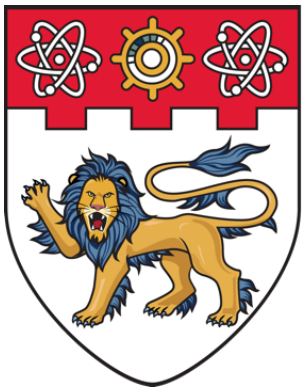


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

ADVANCING ZERO-SHOT AND MULTILINGUAL GENERALIZATION IN LARGE LANGUAGE MODELS

LIU CHAOQUN

Interdisciplinary Graduate Programme

Alibaba-NTU Singapore Joint Research Institute

2026

ADVANCING ZERO-SHOT AND MULTILINGUAL GENERALIZATION IN LARGE LANGUAGE MODELS

LIU CHAOQUN

Interdisciplinary Graduate Programme

Alibaba-NTU Singapore Joint Research Institute

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2026

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

18/08/2025

Date

LIU CHAOQUN

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

18/08/2025

Date

Prof. Luu Anh Tuan

Authorship Attribution Statement

This thesis contains material from 4 papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as **Chaoqun Liu**, Wenxuan Zhang, Guizhen Chen, Xiaobao Wu, Anh Tuan Luu, Chip Hong Chang, and Lidong Bing, "Zero-Shot Text Classification via Self-Supervised Tuning," in Findings of the Association for Computational Linguistics (ACL), 2023.

The contributions of the co-authors are as follows:

- I conceived the central idea of this study, developed the code for training and evaluation, and prepared the initial manuscript.
- Dr. Wenxuan Zhang engaged in regular discussions with me, offered detailed feedback on the research process, and assisted in revising the manuscript.
- Guizhen Chen was responsible for collecting the training data and contributed to the evaluation process.
- Dr. Xiaobao Wu, Prof. Anh Tuan Luu, Prof. Chip Hong Chang, and Dr. Lidong Bing provided valuable feedback on this work and contributed to revising the manuscript.

Chapter 4 is published as **Chaoqun Liu***, Qin Chao*, Wenxuan Zhang, Xiaobao Wu, Boyang Li, Anh Tuan Luu and Lidong Bing, "Zero-to-Strong Generalization: Eliciting Strong Capabilities of Large Language Models Iteratively without Gold Labels," in Proceedings of the International Conference on Computational Linguistics (COLING), 2025.

The contributions of the co-authors are as follows:

- I conceived the central idea of this work, implemented a portion of the code, conducted the majority of the analysis, and prepared the initial manuscript draft.
- Qin Chao developed most of the codebase, performed a portion of the experiments, and contributed to drafting the manuscript.
- Dr. Wenxuan Zhang, Dr. Xiaobao Wu, Prof. Boyang Li, Prof. Anh Tuan Luu, and Dr. Lidong Bing provided valuable feedback on this work and assisted in refining the manuscript.

Chapter 5 is published as **Chaoqun Liu**, Wenxuan Zhang, Yiran Zhao, Anh Tuan Luu and Lidong Bing, "Is Translation All You Need? A Study on Solving Multilingual Tasks with Large Language Models," in Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2025.

The contributions of the co-authors are as follows:

- I conceived the original idea for this work, conducted the majority of the experiments and analyses, and prepared the initial draft of the manuscript.
- Dr. Wenxuan Zhang engaged in regular discussions, offered comprehensive feedback, and assisted in revising the manuscript.
- Yiran Zhao performed analyses on the handling of multilingual prompts by large language models.
- Prof. Anh Tuan Luu and Dr. Lidong Bing provided valuable feedback and contributed to the refinement of the manuscript.

Chapter 6 is published as **Chaoqun Liu**, Wenxuan Zhang, Jiahao Ying, Mahani Aljunied, Anh Tuan Luu and Lidong Bing, "SeaExam and SeaBench: Benchmarking LLMs with Local Multilingual Questions in Southeast Asia," in Findings of the North American Chapter of the Association for Computational Linguistics (NAACL), 2025.

The contributions of the co-authors are as follows:

- I conducted the majority of the experiments and analyses, developed the evaluation scripts used for the benchmarks, and prepared the draft manuscript.
- Dr. Wenxuan Zhang and Jiahao Ying offered insightful feedback and assisted in revising the manuscript.
- Mahani Aljunied annotated the dataset and contributed to the preparation of evaluation prompts.
- Prof. Anh Tuan Luu and Dr. Lidong Bing provided valuable feedback and contributed to the refinement of the paper.

18/08/2025

.....

Date

.....

LIU CHAOQUN

Acknowledgements

I would like to express my gratitude to my supervisors, Prof. Anh Tuan Luu and Prof. Chip Hong Chang, for their invaluable guidance, steadfast support, and insightful feedback throughout the course of my research. I am also grateful to my mentor, Prof. Hui Siu Cheung, for his sustained support. Their encouragement and expertise were instrumental in shaping the direction of my research and in overcoming the challenges encountered along the way.

I extend my gratitude to my Alibaba mentors, Dr. Lidong Bing and Dr. Wenxuan Zhang, for their guidance and encouragement throughout my PhD studies. I also thank my IPP colleagues at Alibaba—including Liying Cheng, Lu Xu, Linlin Liu, Ran Zhou, Bosheng Ding, Qingyu Tan, Chenhui Shen, Yue Deng, Yew Ken Chia, Xingxuan Li, Qin Chao, Guizhen Chen, Donghuizhao Li, and Sicong Leng—for their support and collaboration. I am further grateful to my other colleagues at Alibaba, including Sharifah Mahani Aljunied, Dr. Xin Li, Dr. Xuan-Phi Nguyen, Dr. Jia Guo, Chang Gao, Huiming Wang, Yiran Zhao, Ruochen Zhao, Dr. Houpong Chan, Dr. Hao Zhang, Dr. Weiwen Xu, Dr. Yuming Jiang, and Dr. Gongjie Zhang, among others. I also appreciate the administrative and technical support teams at Alibaba, whose behind-the-scenes efforts ensured a smooth research environment and seamless execution of experiments.

I further extend my gratitude to the staff and researchers at the Alibaba–NTU Singapore Joint Research Institute, including Dr. Xinjia Yu, Kaijun Liu, and Yishu Yin, for their support and encouragement. In addition, I would like to thank the teammates in Prof. Anh Tuan Luu's group, including Xiaobao Wu, Siyue Zhang, and Yandan Zheng, for their friendship and collaboration.

Finally, I extend my heartfelt gratitude to my parents, my brother, and my wife for their unwavering love and support. Their constant care and encouragement sustained my strength whenever I encountered setbacks throughout my PhD journey.

Contents

List of Figures

List of Tables

Abstract

The advancement of Large Language Models (LLMs) has been transformative for natural language processing, yet two fundamental challenges limit their broader impact: the reliance on extensive labeled data for supervised fine-tuning, and the persistent English-centric bias that undermines their effectiveness for the world's linguistic majority. This thesis addresses both challenges through a unified lens of resource efficiency—developing methods that reduce dependence on costly annotations while ensuring equitable model capabilities across languages and cultures.

We first tackle data efficiency by investigating methods that elicit strong model capabilities using solely unlabeled data. We propose Self-Supervised Tuning (SSTuning), a novel paradigm that tunes a language model for zero-shot text classification by learning to predict the first sentence in a paragraph, effectively bridging unlabeled text and downstream tasks without costly annotations. Building on this, we introduce the zero-to-strong generalization framework, an iterative self-annotation process where an LLM progressively unlocks its latent potential on complex classification and reasoning tasks through high-quality pseudo-labels.

We then extend this resource-conscious perspective to multilingualism, where data scarcity is even more acute. We demonstrate that the common "translate-test" strategy, while effective for standard NLP tasks, fails on culture-related queries where native language prompting proves essential—exposing how English-centric evaluation shortcuts mask true multilingual capabilities. To address this, we develop SeaExam and SeaBench, benchmarks constructed from authentic Southeast Asian educational and conversational scenarios that more accurately assess regional language performance.

In summary, this thesis advances a coherent vision for developing more accessible and equitable language technologies: first by reducing the data requirements that bottleneck model development, and then by ensuring that evaluation practices do not perpetuate linguistic and cultural biases.

Chapter 1

Introduction

1.1 Motivation

Recent advances in pre-trained language models (PLMs) have revolutionized the field of natural language processing (NLP) [4, 17], yet their effectiveness is often contingent on vast amounts of labeled data. This data dependency creates a significant bottleneck, making it costly and impractical to train or fine-tune models for every new task. This challenge has fueled interest in paradigms like zero-shot learning, which has attracted considerable research attention for its ability to conduct inference on unseen tasks without specific training data [3, 18, 19].

However, existing zero-shot methods present their own limitations. One popular approach, prompting, is notoriously sensitive to the design of templates and verbalizers [2], making it difficult to generalize across tasks without extensive manual engineering. A second approach, meta-tuning, mitigates some of these issues by fine-tuning a PLM on a collection of related, labeled tasks [3, 19, 20]. Yet, this still presupposes the availability of large-scale annotated data, narrowing its application scope and failing to solve the core data-dependency problem. This gap motivates our initial research question: How can we leverage the intrinsic structure of unlabeled text, much like in pre-training [4, 21], to perform robust zero-shot learning? To address this, we propose Self-Supervised Tuning (SSTuning), a novel approach that exploits self-supervised signals at the tuning stage to enable zero-shot classification without relying on human annotations or brittle templates. SSTuning trains a language model to predict which first sentence belongs to a given paragraph, using

first sentences from other paragraphs—especially those from the same article—as hard negatives. This objective teaches the model to match texts with their summarizing "labels" based on semantics rather than superficial keyword overlap. At inference, the tuned model can directly classify unseen texts by selecting the most relevant verbalizer from candidate labels, requiring no further fine-tuning or labeled data.

Building on this, we consider an even more challenging scenario where tasks are too complex for humans to provide reliable gold-standard labels. This connects to the weak-to-strong generalization paradigm, where, as shown by Burns et al. [8], a stronger model can be effectively supervised by a weaker one. However, this approach is still constrained by the supervisor's capabilities and requires a pre-existing supervisor model. This limitation leads to a more fundamental question: Can a large language model achieve strong performance without any external supervision at all? This is inspired by findings that models can learn effectively even from random labels [22, 23] or invalid reasoning paths [24]. This motivates the development of a new framework we term zero-to-strong generalization. This framework bootstraps LLM performance without any gold labels by initially prompting the model with random or invalid demonstrations, then iteratively selecting higher-confidence predictions as new demonstrations. This self-reinforcing process progressively improves label quality across iterations, enabling the model to unlock its latent capabilities on classification and reasoning tasks. Experiments show this approach can match or even surpass in-context learning with gold labels, with stronger effects observed for more capable models and complex tasks.

We then extend this resource-conscious perspective to multilingualism, where data scarcity is even more acute for non-English languages. While LLMs demonstrate multilingual capabilities, their training corpora are overwhelmingly English-centric [9, 25, 26], which can lead to suboptimal performance in other languages [27–29]. A common strategy to circumvent this is to translate non-English queries into English (the "translate-test" method), a technique that has been applied at both the training and inference stages [10, 30–32]. Despite its apparent effectiveness, this approach has been underexplored for real-world user queries. We hypothesize that this strategy is a brittle workaround that fails on tasks requiring deep cultural and linguistic nuance, motivating a comprehensive analysis to understand its true limitations.

This investigation, in turn, exposes a deeper, more systemic issue: the inadequacy of current multilingual evaluation benchmarks. Many prominent benchmarks, such as Multilingual MMLU, MGSM, and XNLI, are constructed by simply translating existing English datasets [13, 16, 33, 34]. As has been argued, this practice fails to capture the unique cultural contexts and practical applications of the target language [35]. A translated question about an American landmark, for instance, is a poor instrument for assessing a model's understanding of Southeast Asian culture. This significant divergence between translated content and authentic local queries means we are not accurately measuring the true multilingual capabilities of LLMs. This work provides the first comprehensive analysis of how translation affects LLM performance across both standard NLP tasks and real user queries containing culture-specific knowledge. While translate-test achieves strong results on conventional multilingual benchmarks, native language prompting proves more effective for culture-related queries, particularly with advanced and non-English-centric LLMs. These findings reveal that translation introduces a trade-off: it may boost task performance but risks losing cultural and linguistic nuances that only native prompting can capture.

This critical gap in evaluation motivates the final contribution of this thesis. Following the design principles of widely-used benchmarks like MMLU [16] and MT-Bench [36], but focusing on authentic content, we introduce SeaExam and SeaBench: two novel benchmarks built from the ground up using real-world materials from Southeast Asia. By providing culturally and contextually relevant evaluation tools, we can more effectively discern model capabilities, identify weaknesses, and ultimately drive the development of more globally competent and equitable language models.

While our research develops methods to reduce labeled data dependency for training, we simultaneously invest in constructing new evaluation benchmarks—an apparent contradiction. However, data efficiency in learning and data quality in evaluation serve complementary roles. Self-supervised methods become valuable precisely because they scale to low-resource languages, but verifying this requires benchmarks that do not reward English-centric shortcuts. Our training methods address the quantity bottleneck, while SeaExam and SeaBench address the quality bottleneck, together advancing the shared goal of equitable multilingual model development.

1.2 Contributions

The main contributions of the thesis are:

- We propose **Self-Supervised Tuning (SSTuning)**, a new learning paradigm to solve zero-shot text classification tasks using only unlabeled data. This framework uses a simple yet effective learning objective, First Sentence Prediction, to bridge the gap between self-supervision and downstream tasks. Extensive experiments on 10 datasets show that SSTuning achieves state-of-the-art accuracy in both topic classification and sentiment analysis.
- We introduce the ***zero-to-strong generalization*** framework, a simple and effective method to elicit the strong capabilities of LLMs iteratively without requiring any gold-standard labels. We demonstrate its effectiveness across 21 diverse classification and reasoning tasks and provide an analysis of its underlying principles, confirming that its benefits extend to both fine-tuning and larger model sizes.
- We conduct a comprehensive empirical study on multilingual prompting strategies, which finds that translation remains a strong baseline but is not universally optimal. By expanding the evaluation to include real-world user queries and non-English-centric models, we expose critical gaps in current multilingual evaluation and underscore the need for more comprehensive benchmarks.
- We address the identified evaluation gaps by introducing two novel, culturally-grounded benchmarks, **SeaExam** and **SeaBench**. Designed for the Southeast Asian (SEA) context, we show that these benchmarks have a closer distribution to real-world queries and enable a more effective and accurate differentiation of model performance compared to their translated counterparts.

1.3 Organization of the report

The rest of this report is organized as follows:

- Chapter 2 reviews the existing works for low-resource methods for NLP and multilingual large language models.
- Chapter 3 presents the proposed method for zero-shot text classification via self-supervised tuning.
- Chapter 4 introduces zero-to-strong generalization, which elicits strong capabilities of large language models iteratively without gold labels.
- Chapter 5 demonstrates that while translation into English can boost the performance of English-centric LLMs on NLP tasks, it is not universally optimal.
- Chapter 6 introduces two novel benchmarks, SeaExam and SeaBench, designed to evaluate the capabilities of Large Language Models (LLMs) in Southeast Asian (SEA) application scenarios.
- Chapter 7 concludes the report and discusses future work.

Chapter 2

Background and Literature Review

In this chapter, we review related works and preliminaries to facilitate understanding of the subsequent chapters. First, we review learning methods used to address problems in low-resource scenarios, such as those with limited labeled data. Subsequently, we address challenges in multilingual contexts, encompassing prompting strategies and benchmarking methodologies.

2.1 Low-Resource Methods for NLP

2.1.1 Zero-shot Learning

Previous zero-shot learning methods can be broadly categorized into two types: prompting-based methods and meta-tuning-based methods.

2.1.1.1 Prompting-based methods

Zero-shot learning has the advantage that no annotated data is required for downstream tasks but it is a challenging task even for humans since no demonstration is allowed. One solution to this challenge is to increase the model parameters. Large

Language Models (LLM), e.g., GPT-3 [1] and PaLM [37], achieved promising performances in zero-shot settings by using prompting. As shown in Figure 2.1, LLMs can do zero-shot inference with a task description and a prompt. However, LLMs are not usable in many real-world scenarios due to their surprisingly large sizes.

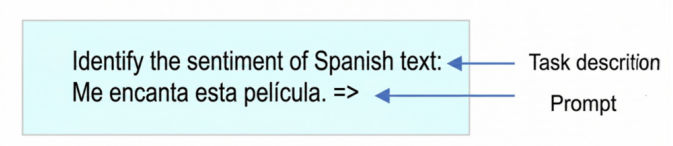


FIGURE 2.1: Zero-shot learning with prompt: inference on unseen tasks without fine-tuning [1].

A promising alternative is to reformulate tasks like text classification as cloze questions with a pattern and a verbalizer [38, 39], making small language models capable of zero-shot learning. Even though such methods make it possible for zero-shot learning, prompting is known to be sensitive to the templates, which may be designed in different formats [40]. To eliminate the need for labeled training data, Meng et al. [40] proposed a supervision generation approach. By generating labeled data with a generative PLM, followed by data filtering and fine-tuning, the trained model can be applied to the target task. Alternatively, van de Kar et al. [2] create labeled datasets by mining labeled data samples from an unlabeled corpus, after filtering and fine-tuning, which can perform better than prompting in zero-shot settings, as shown in Figure 2.2. Such methods avoided using labeled data but have the constraint that a specific model needs to be fine-tuned for each downstream task, thus not efficient for deployment.

2.1.1.2 Meta-tuning based methods

To enhance the zero-shot capabilities of PLMs, a bunch of supervised tuning methods have been proposed. Instruction-tuning-based models like FLAN [3] and T0 [18], fine-tine PLMs on a collection of datasets described by instructions or prompts to improve performance on unseen tasks. As shown in Figure 2.3, by fine-tuning the model on tasks like commonsense reasoning, translation, and sentiment analysis tasks, FLAN can inference on unseen tasks like natural language inference directly.

UnifiedQA [20] formats multiple tasks as question answering (QA) format. After fine-tinging on a collection of tasks, can perform well on unseen tasks. UnifiedQA

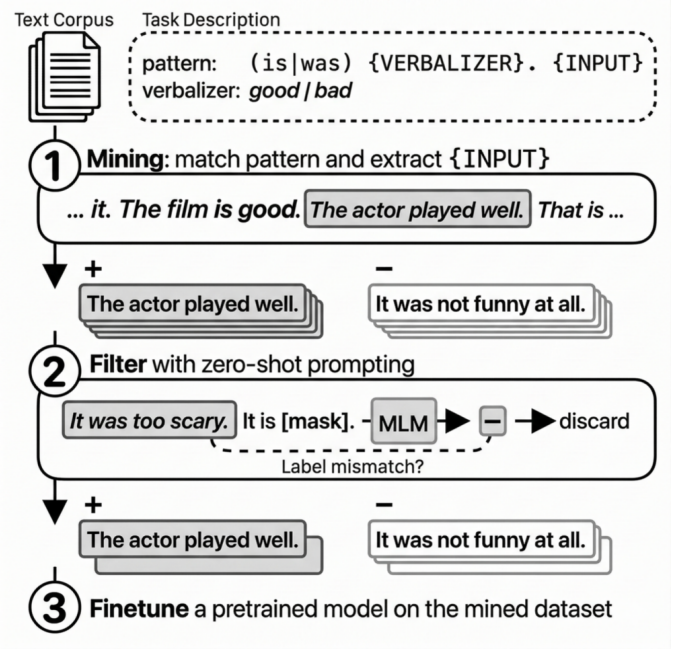


FIGURE 2.2: Mining-based method [2]: 1) Mine examples with labels from a corpus with regex-based patterns; 2) Filter examples that are predicted to have a different label with zero-shot prompting; 3) Fine-tune a PLM by adding a classification head.

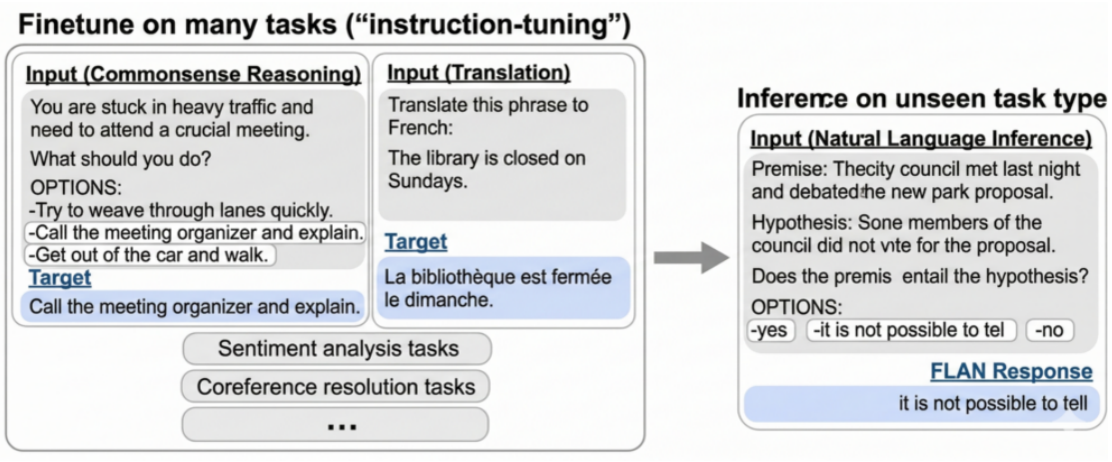


FIGURE 2.3: Instruction tuning in FLAN [3].

can also be further meta-tuned [41] on text classification datasets and do zero-shot on other classification datasets. UniMC [19] converts zero-shot learning on multiple-choice tasks and do zero-shot inference on tasks that can be formulated in the same format. Another line of work is to convert text classification problems to textual entailment problems. By fine-tuning on natural language inference datasets [42] or a dataset from Wikiepdia [43], the models can do inference directly on text classification datasets. All of the methods share the common features that they

use labeled datasets.

2.1.2 Self-supervised Learning

2.1.2.1 Token Level

Self-supervised learning has been widely applied during language model pre-training by leveraging the input data itself as supervision signals [44]. Left-to-right language modeling [17, 45] and masked language modeling [4, 21, 46] help learn good sentence representations. Left-to-right models or auto-regressive models maximize the probability of generating the next token based on previsou tokens. For masked language modeling like BERT, some input tokens are masked randomly and then predicted during pre-training, as shown in Figure 2.4. Following BERT, SpanBERT [47] masks random contiguous random spans and predict the entire span with the span boundary representations.

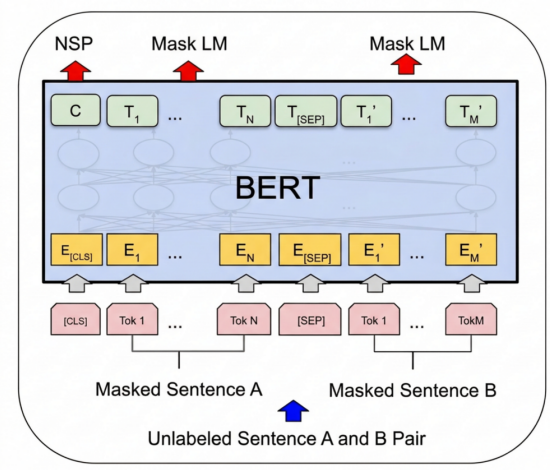
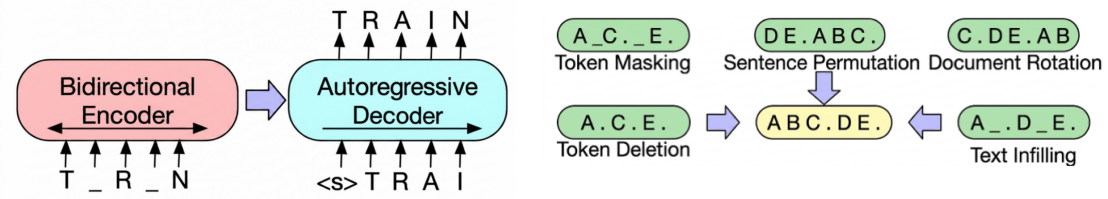


FIGURE 2.4: Pre-training for BERT [4].

Unlike encoder-based models, which can only predict the same number of tokens in the output as input, encoder-decoder-based models can generate more flexible outputs. Lewis et al. [5] proposed a denoising autoencoder called BART that can learn a model to reconstruct the original text from the corrupting text. As shown in Figure 2.5, the transformation functions include token masking, sentence permutation, document rotation, token deletion, and text infilling. During fine-tuning, both the encoder and decoder use the uncorrupted document as input.



(A) Input and output format (B) Input transformations

FIGURE 2.5: BART input and output formulations [5].

With the encoder-decoder architecture, the model does not need to reconstruct the whole original text, but can also generate the masked text span. As shown in Figure 2.6, T5 [6] replace consecutive spans of tokens with sentinel tokens (shown as <X> and <Y>). The resulting sequence consists of dropped spans, surrounded by special tokens that were used to replace them in the original sequence, and ending with a final special token <Z>.

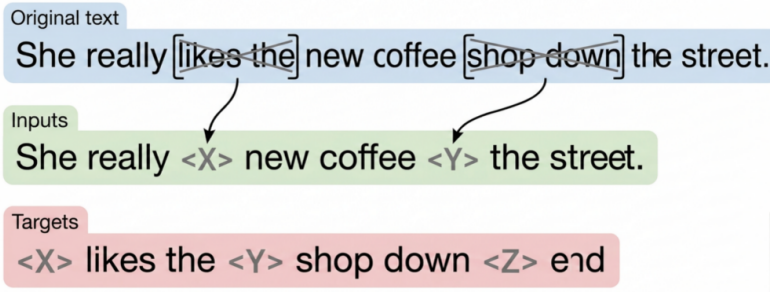


FIGURE 2.6: T5 input and output formulations [6].

2.1.2.2 Sentence Level

In order to capture the sentence-level relations of downstream tasks, Devlin et al. [4] pre-train a next sentence prediction (NSP) task, which is to predict whether S_2 is the next sentence that follows S_1, given a sentence pair (S_1, S_2) as input, as shown in Figure 2.4. Lan et al. [21] use sentence order prediction task to model the inter-sentence coherence. Wang et al. [7] combine the two objectives to form a three-way classification task, which can predict whether S_2 is a sentence that follows S_1, a sentence that precedes S_1, or a sentence randomly sampled from another document, as shown in Figure 2.7.

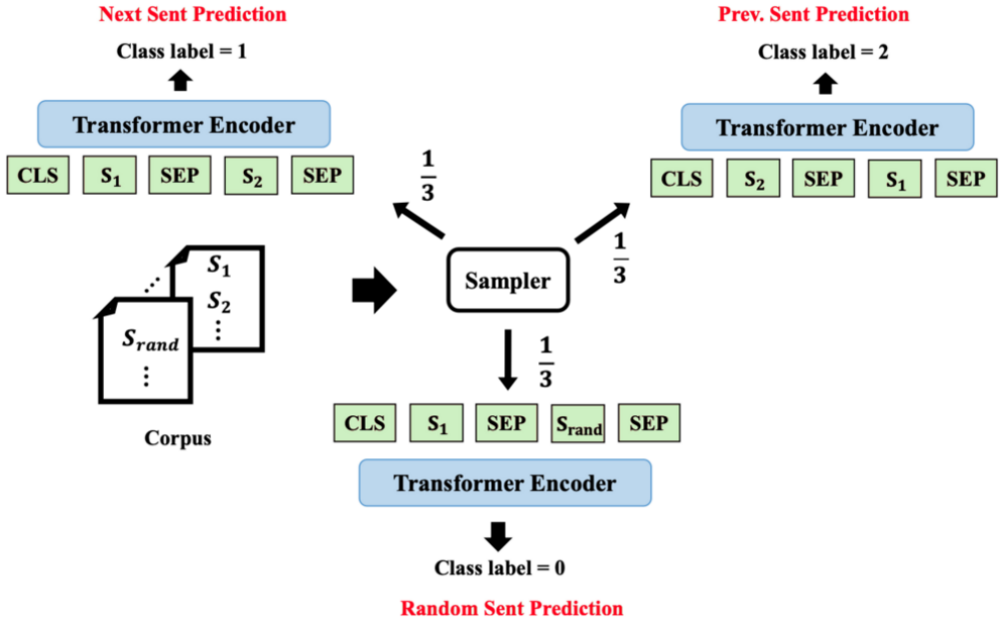


FIGURE 2.7: Sentence structural objective in StructBERT [7].

Instead of modeling the inter-sentence relations, Meng et al. [48] employ sequence contrastive learning to align the corrupted text sequences that originate from the same input source and guarantee the uniformity of the representation space.

2.1.3 Weak-to-Strong Generalization.

In the future, advanced models will handle complex tasks with only weak human supervision. To study this, Burns et al. [8] proposed using weak supervisor models to elicit the capabilities of stronger student models, as illustrated in Figure 2.8. Their findings revealed that, after fine-tuning, the strong student models consistently outperformed the weak supervisor models, a phenomenon they term *weak-to-strong generalization*. In contrast to transferring knowledge from strong models to weak models [49, 50], this learning paradigm is a specific type of weakly-supervised learning [51], where models are trained with noisy or biased labels [52–56]. Our work eliminates the necessity of weak models or weak labels for supervision. Instead, we utilize minimal supervision, such as the label space or incorrect initial demonstrations, to elicit the capabilities of large language models. Other research has proposed self-improvement of LLMs using labeled or unlabeled data [57–59] for reasoning tasks. In contrast, we aim to propose a general framework for learning new tasks without labeled data.

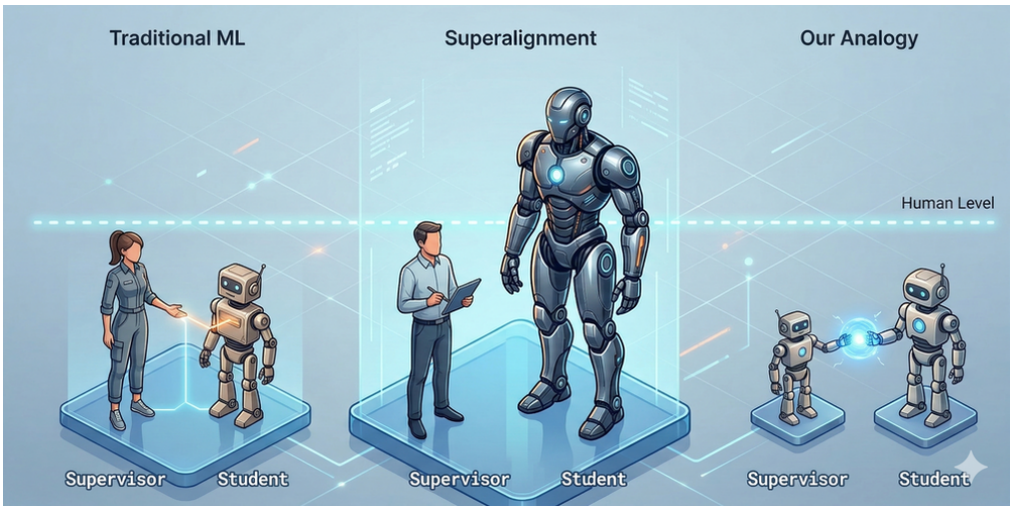


FIGURE 2.8: Comparison of traditional ML, superalignment, and weak-to-strong analogy [8].

2.1.4 In-context Learning

In-context learning (ICL) [9] can effectively learn new tasks with a few demonstrations, but its mechanism is still under discussion. As shown n Figure 2.9, a task description and some examples are provided for the model to learn new tasks. Previous research [60–62] found that ICL is sensitive to the demonstration samples, their order, and their diversity. Studies by Min et al. [22] and Wang et al. [24] discovered that even random labels for classification or invalid demonstrations for reasoning tasks can yield good performance, suggesting that gold labels are not always necessary. However, Yoo et al. [23] showed that correct input-label mappings can have varying impacts through extensive experiments. Recently, Wang et al. [63] found that learning to retrieve in-context examples helps improve the performance, but the gold labels are needed.

2.2 Mutilingual Large Language Models

2.2.1 Multilingual Prompting Strategies

The translate-test is a popular technique used to refine the performance of multilingual NLP benchmarks [30, 31, 34, 64–66]. In the era of LLMs, various strategies have been developed to enhance the performance of LLMs using multilingual

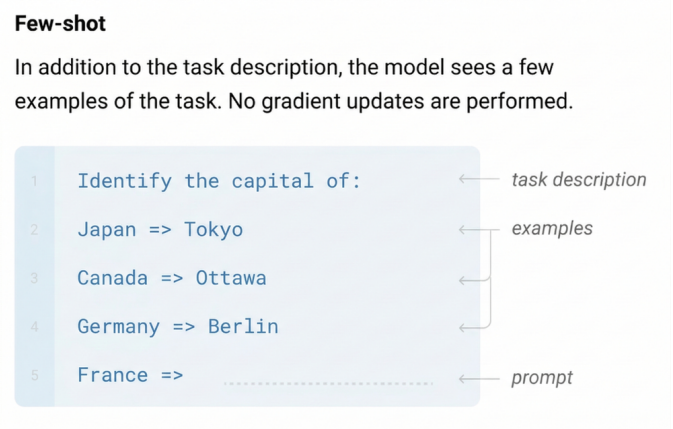


FIGURE 2.9: An example of in-context learning [9].

datasets. Shi et al. [13] discovered that EN-CoT outperforms NATIVE-CoT. Huang et al. [10] introduced cross-lingual-thought prompting (XLT) to minimize language disparities. As shown in Figure 2.10, to generate responses in the desired format, a language-independent prompt is created by populating an XLT template with request metadata, which is then sent to the LLM. In parallel, Qin et al. [32] introduced cross-lingual prompting, and Etxaniz et al. [67] suggested self-translate to elevate their performances. Effective in translating prompts into English, these methods excel in NLP tasks but remain uncertain in real-world applications. Their success hinges on the English-centric nature of the LLMs. Our study evaluates translation effectiveness across NLP tasks, real user queries, and non-English-centric LLMs, revealing the limitations of these methods.

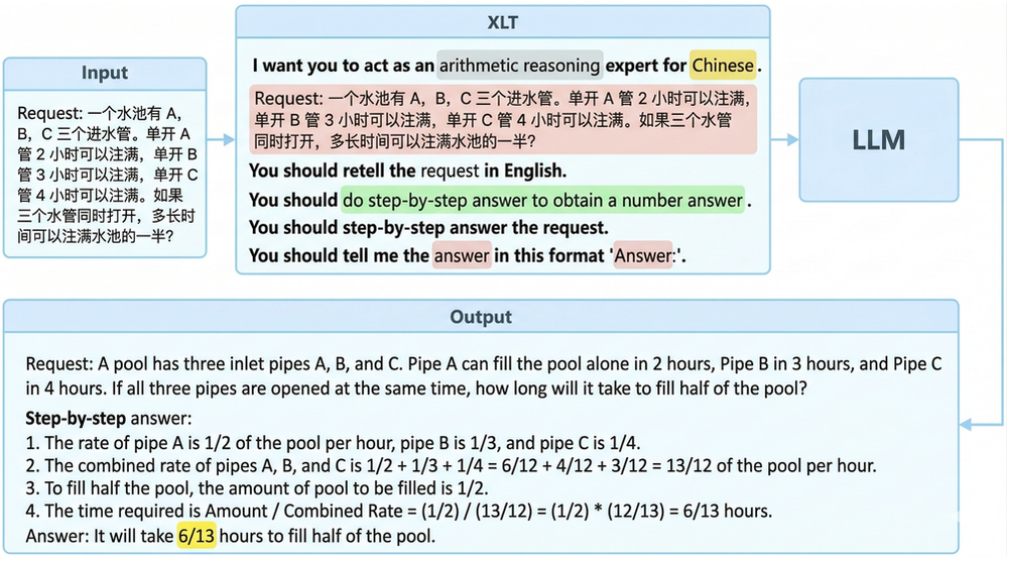


FIGURE 2.10: Overview of cross-lingual-thought prompting [10].

2.2.2 Multilingual Evaluation

Since the release of ChatGPT, the evaluation of LLMs has attracted the attention of the research community [68, 69]. Shi et al. [13] evaluated LLMs on MGSM and found that the models demonstrated strong multilingual reasoning capabilities, even for low-resource languages. They found that the same model perform differently on different languages or with different prompting strategies. Bang et al. [69] evaluated ChatGPT on 23 datasets covering 8 NLP tasks. They found that ChatGPT failed to generalize its capabilities to non-Latin scripts. To cover tasks, Ahuja et al. [27] evaluated ChatGPT and GPT-4 on 16 NLP datasets across 70 languages and compared them with state-of-the-art non-autoregressive models. Concurrently, Lai et al. [28] evaluated ChatGPT on 7 different tasks across 37 diverse languages. However, these evaluations are primarily limited to standard NLP tasks and largely overlook real-world scenarios and cultural knowledge [70], which are crucial for understanding the practical applicability of LLMs.

2.2.3 LLM-as-a-Judge

Strong LLMs have emerged as judges to evaluate model capabilities on open-ended questions. Zheng et al. [36] proposed MT-bench, with GPT-4 as the judge to test multi-turn conversation and instruction-following ability. Li et al. [71] introduced AlpacaEval, a method for assessing a model's performance by determining the percentage of instances in which a powerful LLM favors the model's outputs compared to those from a reference model. Building on this, Dubois et al. [72] proposed length-controlled AlpacaEval to mitigate length gameability, as judge LLMs prefer longer outputs. To effectively distinguish model capabilities and capture human preferences in practical scenarios, Li et al. [73] developed Arena-Hard, a data pipeline designed to create high-quality benchmarks using live data from Chatbot Arena [36]. Similarly, Lin et al. [74] proposed Wildbench to benchmark LLMs with real user queries. These benchmarks are limited to use LLMs as English judges. Hada et al. [75] expand the evaluation of LLM-based evaluators to eight languages, but not including SEA languages. To our knowledge, SeaBench is the first open-ended multi-turn benchmark for SEA languages.

2.2.4 SEA Benchmarks

Several benchmarks have been developed to evaluate LLMs on SEA languages. SeaEval [76] includes 28 datasets covering classic NLP tasks, reasoning, and cultural comprehension. For the newly created datasets, Cross-MMLU and Cross-LogiQA, the questions were translated from English using Google Translate and proofread by native speakers. SeaCrowd benchmarks [77] cover 4 NLU tasks with 131 data subsets and 7 NLG tasks with 100 subsets. BHASA [78] offers a holistic evaluation suite for assessing linguistic and cultural aspects in LLMs tailored to SEA languages. These benchmarks aim to provide a comprehensive evaluation for SEA languages, with a focus on NLP tasks. However, none of the existing benchmarks evaluate open-ended questions or multi-turn conversations. In contrast, SeaExam focuses on real-world exam questions, and SeaBench offers the first SEA benchmark designed specifically for open-ended and multi-turn evaluations.

Chapter 3

Zero-Shot Text Classification via Self-Supervised Tuning

3.1 Introduction

Recent advances in pre-trained language models (PLMs) have brought enormous performance improvements in a large variety of NLP tasks [4, 17]. These paradigm shifts towards leveraging generic features learnt by PLMs are driven by the high data cost required for learning each new NLP task afresh. One promising learning method that echoes this paradigm shift is zero-shot text classification, which predicts text labels on unseen tasks. Zero-shot text classification has attracted considerable research attention in recent years [3, 18, 19], as labeled data is no longer a necessity for relearning new feature representations for untrained specific tasks.

Existing studies on zero-shot text classification can be briefly classified into two types, as shown in Figure 3.1. The first type is prompting, which uses PLMs to predict labels with designed templates and verbalizers (Figure 3.1 (a)). This can be achieved by leveraging the generation capability of large language models [1, 37], or reformulating text classification tasks as mask-filling tasks [38, 39]. Likewise, generation-based methods [40, 79] and mining-based methods [2] also rely on prompting to generate or filter noisy labeled samples, which are used for further fine-tuning. The second type is meta-tuning which fine-tunes a PLM on a collection of labeled data of related tasks before conducting inference on unseen tasks

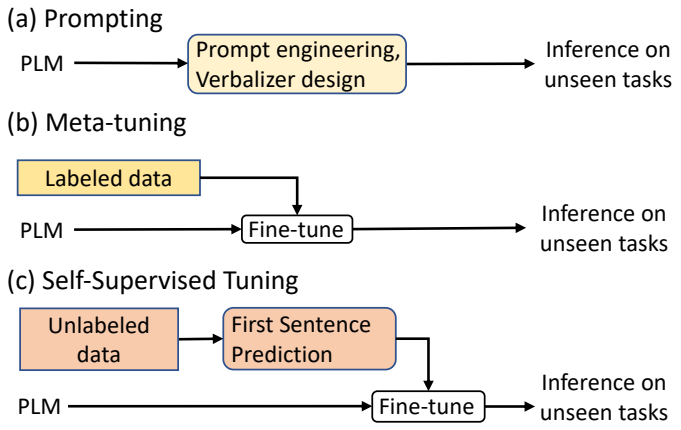


FIGURE 3.1: Zero-shot learning approaches: (a) prompting, (b) meta-tuning, and (c) our proposed self-supervised tuning method.

(Figure 3.1 (b)). By reformulating the annotated data into instruction templates [3, 18], question-answer pairs [20, 41], multiple-choice questions [19] or entailment pairs [42, 43, 80], and fine-tuning on them, PLMs perform well on unseen tasks.

Despite the achieved performance, existing methods have several limitations for wider applications. Prompting has shown to be sensitive to the choice of patterns and verbalizers [2]. This makes it difficult to design different templates specifically for each task. In addition, generation-based and mining-based methods require fine-tuning PLMs for each downstream task, which is inefficient for deployment. On the other hand, meta-tuning relies on labeled data of relevant tasks or in specific formats to facilitate the learning of desired patterns. The requirement for such large-scale annotated data narrows its application scope.

To address the above issues, we propose to leverage self-supervised learning (SSL) for zero-shot text classification tasks. SSL has been widely used during the pre-training stage of PLMs to alleviate the need for large-scale human annotations [4, 21] by exploiting the intrinsic structure of free texts. Therefore, with a suitable SSL objective, the model is able to capture certain patterns with the auto-constructed training data and can be applied to a wide range of downstream tasks in a zero-shot manner without specific designs. To our best knowledge, this is the first work to exploit SSL at the tuning stage for zero-shot classification, which we refer to as self-supervised tuning (SSTuning).

The biggest challenge of applying self-supervised learning to zero-shot text classification tasks is to design a proper learning objective that can effectively construct

large-scale training samples without manual annotations. Intuitively, the core of the text classification task can be treated as associating the most suitable label to the text, given all possible options. Motivated by this observation, we propose a new learning objective named first sentence prediction (FSP) for the SSTuning framework to capture such patterns. In general, the first sentence tends to summarize the main idea of a paragraph. Therefore, predicting the first sentence with the rest of the paragraph encourages the model to learn the matching relation between a text and its main idea ("label"). Even when the first sentence does not encapsulate the entire paragraph, it typically maintains a stronger semantic relationship with the following sentences compared to other paragraphs. To generate training samples, we use the first sentence in the paragraph as the positive option and the rest as text. The first sentences in other paragraphs are used as negative options. Specifically, if negative options are from the same article as the positive option, we call it hard negatives since the sentences in the same article normally have some similarities, such as describing the same topic. Hard negatives force the model to learn the semantics of the text instead of simply matching the keywords to complete the task.

In the inference phase, we convert all possible labels of a sample into verbalizers as options. The tuned model can thus retrieve the most relevant option as the predicted label of the text. Since the tuned model has seen a large number of samples and various first sentences as options, which has a higher chance to consist of similar options to the ones at the inference phase, it is easier and more flexible to design a proper verbalizer. In this way, our SSTuning enables efficient deployment of PLM for classifying texts of unseen classes on-the-fly without requiring further fine-tuning with labeled data or unlabeled in-domain data.

Our main contributions are:

- We propose a new learning paradigm called self-supervised tuning (SSTuning) to solve zero-shot text classification tasks. A simple yet effective learning objective named first sentence prediction is designed to bridge the gap between unlabeled data and text classification tasks.
- We conduct extensive experiments on 10 zero-shot text classification datasets. The results show that SSTuning outperforms all previous methods on overall accuracy in both topic classification tasks and sentiment analysis tasks.

3.2 Proposed Method

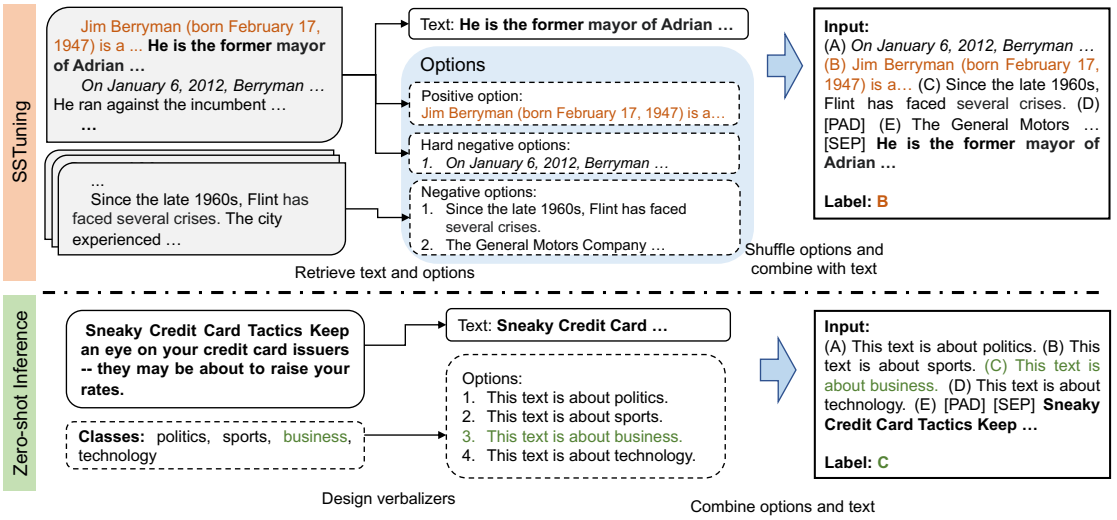


FIGURE 3.2: Data construction for SSTuning (top) and zero-shot inference (bottom). The number of labels N_{model} is set as 5 here. The SSTuning example is from Wikipedia and the inference example is from AG News dataset.

In this section, we discuss our proposed framework, SSTuning, and provide details for our dataset preparation process using the idea of first sentence prediction (FSP), the tuning phase, and the zero-shot inference phase.

3.2.1 First Sentence Prediction

Text classification can be regarded as selecting the most relevant label for the text, given all possible labels. Based on such observation, we propose the FSP task to create datasets for our SSTuning by mimicking the same structure.

We design the FSP task by considering both the nature of the unlabeled corpus and the input/output format of classification tasks. In this subsection, we describe in detail how to construct the tuning and validation sets from the unlabeled corpus. Figure 3.2 shows the core procedures for our dataset generation.

Data filtering. We first filter data to select appropriate paragraphs for tuning. The original unlabeled datasets can be noisy and some paragraphs are not suitable for generating tuning datasets. We filter the paragraphs with the following features: 1) the paragraph only contains 1 sentence; 2) the first sentence contains less than or

equal to 3 characters; 3) the first sentence only contains non-alphabetic symbols; 4) repeated paragraphs. Removing meaningless sentences ensures data quality, which helps improve the performance of the model.

First sentence as the positive option. We consider an article A_n that contains M paragraphs, i.e., $A_n = [P_1^n, P_2^n, ...P_M^n]$, and suppose paragraph P_m^n has K sentences $[S_1^{n,m}, S_2^{n,m}, ..., S_K^{n,m}]$, the positive option $O_c^{n,m}$ and the text $x^{n,m}$ are:

$$O_c^{n,m} = S_1^{n,m} \tag{3.1}$$

$$x^{n,m} = [S_2^{n,m}, ..., S_K^{n,m}] \tag{3.2}$$

As shown in Figure 3.2, we can retrieve the first sentence *"Jim Berryman (born February 17, 1947) is a ... "* as the positive option and the rest of the paragraph *"He is the former mayor of Adrian ..."* as the text for the first paragraph in the article.

Negative sampling. After getting the positive option, we randomly sample J "first sentences" from other paragraphs $[S_1^{n_1,m_1}, S_1^{n_2,m_2}, ...S_1^{n_J,m_J}]$ as negative options, where J is a random number that satisfies $1 \leq J \leq N_{\text{maxLabel}} - 1$. We let N_{maxLabel} denote the maximum number of labels that are first sentences, which is pre-defined to ensure the total number of tokens for options is not too long. It is less or equal to N_{model}, where N_{model} is the number of labels for the model output layer. Having a random number of negative options bridges the gap between tuning and zero-shot inference since the number of classes for evaluation datasets may vary from 2 to N_{model}.

Hard negatives. During negative sampling, if the negative options and the positive option are from the same paragraph ($n_j = n$), we call the options hard negatives. Inspired by the successful application of hard negatives in Gao et al. [81], we purposely add more hard negatives to enhance the model performance. In some articles, we observe that in the same paragraph, common words are likely to appear in the first sentence and the rest of the paragraph at the same time. As shown in Figure 3.2, *"Berryman"* can be a shortcut to select the corresponding first sentence for the text. However, if we add the hard negative *"On January 6,*

2012, Berryman ...”, the model needs to understand the true semantics to select the positive option.

Option padding. We pad the options with the special ”`[PAD]`” token to make the input format consistent between the tuning phase and the inference phase. Specifically, if the total number of options after negative sampling is $(J+1) < N_{\text{model}}$, we will add $(N_{\text{model}} - J - 1)$ `[PAD]` options. Thus the final list of options is:

$$\begin{aligned} O^{n,m} = [&S_1^{n,m}, S_1^{n_1,m_1}, S_1^{n_2,m_2}, ...S_1^{n_J,m_J}, \\ &O^1_{\texttt{PAD}}, O^2_{\texttt{PAD}}, ...O^{N_{\text{model}}-J-1}_{\texttt{PAD}}] \end{aligned} \tag{3.3}$$

Generating final text and label. We shuffle the option list because the position of a positive option is random in the evaluation datasets. After shuffling, we assume the option list is:

$$O^{n,m}_{\text{shuffle}} = [O_0, O_1, ...O_{N_{\text{model}}-1}], \tag{3.4}$$

where the positive option $O^{n,m}_c = O_j$. Then the label for this sample is:

$$L^{n,m} = j. \tag{3.5}$$

The final input text is the concatenation of the above components:

$$x^{n,m}_{inp} = [\text{CLS}]\{(T_i)\ O_i\}_{i=0}^{N_{\text{model}}-1}[\text{SEP}]x^{n,m}[\text{SEP}] \tag{3.6}$$

where T_i is the i-th item from the index indicator list T (e.g. $[A, B, C...]$), `[CLS]` is the classification token, and `[SEP]` is the seperator token used by Devlin et al. [4].

Thus the final text-label pair ($x^{n,m}_{inp}$,$L^{n,m}$) is the generated sample. We can repeat this process to generate a large number of samples as the tuning set. The validation set can also be generated in the same way. Note that if we select a corpus that only contains paragraphs instead of articles, we can treat each paragraph as an article, and no hard negatives are generated.

3.2.2 Tuning Phase

3.2.2.1 Network Archtecture

We employ BERT-like pre-trained masked language models (PMLM) as the backbone, such as RoBERTa [46] and ALBERT [21]. Following Devlin et al. [4], we add an output layer for classification. Such models have both bidirectional encoding capabilities and simplicity. Generative models are not necessary since we only need to predict the index of the correct option. We do not make any changes to the backbone so that the method can be easily adapted to different backbones. In order to cover all test datasets, we config the number of labels for the output layer as the maximum number of classes for all test datasets, denoted by N_{model}.

3.2.2.2 Learning Objective

Traditional text classification with PMLMs like BERT maps each classification layer output to a class. Such design requires a dedicated output layer for each dataset as they have different classes. Instead, our learning object for FSP with the same network is to predict the index of the positive option. In this way, we can use the output layer for both tuning and inference, and for all various kinds of datasets.

As shown in Figure 3.2, we concatenate the labels and the text as input. The outputs are the indices (0, 1, 2..., which correspond to A, B, C), which are the same as traditional classification datasets. We use a cross-entropy loss for tuning the model.

3.2.3 Zero-Shot Inference Phase

During the zero-shot inference phase, we can infer directly by converting the input of the sample to the same format as that in the tuning phase.

3.2.3.1 Input Formulation

As shown in Figure 3.2, the zero-shot inputs are formulated similarly as the tuning phase, except 1) instead of using first sentences as options, we convert the class names to verbalizers as options. 2) No shuffling is needed. Since the converted input and output during SSTuning and zero-shot phases are the same, no further adjustment of the model is required.

3.2.3.2 Constrained Prediction

Since the dimension of the output logits (N_{model}) may be different from the number of classes in a dataset (N_L), the predictions may be out of range (e.g. the model may output 3 for a dataset with 2 classes). To solve this issue, we simply make predictions based on the first N_L logits:

$$P = \text{argmax}(\text{logits}[0 : N_L]) \tag{3.7}$$

where P is the index for the positive option.

3.3 Experiment Setup

3.3.1 SSTuning Datasets

We choose English Wikipedia and Amazon review dataset (2018) [82] for SSTuning. The two datasets are large: the Wikipedia corpus has more than 6.2M articles[1] by the end of 2021, while Amazon Review Data has around 233.1M reviews[2]. Wikipedia articles typically use formal expressions and Amazon reviews contain informal user-written texts, together covering different genres of text.

For English Wikipedia, we collect articles up to March 1st, 2022. To balance the dataset, we select up to 5 paragraphs in each article. The generated dataset has 13.5M samples. For the Amazon review dataset, we only use the review text to create our SSTuning dataset, ignoring other information such as summary and

[1] `https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia`
[2] `https://nijianmo.github.io/amazon/`

vote. The Amazon review dataset has 29 categories. To keep the model from being dominated by a certain category, we select up to 500k samples from each category. In the end, we collected 11.9M samples.

To have a balanced dataset, we sample 2.56M from the Wikipedia dataset and 2.56M from the Amazon review dataset, forming a total of 5.12M samples as the tuning dataset. In addition, we sampled 32k from each of the two datasets, forming a validation set consisting of 64k samples. Some of the final generated samples from English Wikipedia and Amazon product reviews are shown in Table 3.1.

Dataset	**Label**	**Positive Option**	**Generated Text**
Wikipedia	0 (A)	Rawat emigrated to Canada from India in 1968.	(A) Rawat emigrated to Canada from India in 1968. (B) Meskowski was a racing car constructor. (C) [PAD] (D) , there were 42 people who were single and never married in the municipality. (E) [PAD] (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) It is a Church of England school within the Diocese of Salisbury. (L) Falkoner Allé was opened to the public after Hømarken (literally " Hayfield "), an area to the north belonging to Ladegården, originally a farm under Copenhagen Castle, was auctioned off. (M) [PAD] (N) [PAD] (O) In the fall of her senior year at McDonogh, Cummings committed to play for the University of Marylandś womenś lacrosse team as the nationś top recruit. (P) Ranville is a native of Flint, Michigan and attended St. Agnes High School. (Q) The Dodgeś Institute of Telegraphy was housed in the Institutes building at 89 East Monroe. (R) During 2004 - 2011, Rawat was President of the Communications Research Centre, Canadaś centre of excellence for telecommunications R & D, with 400 staff and an annual budget of over $ 50 million. (S) [PAD] (T) [PAD] [SEP] She speaks English, French, Hindi and Spanish.
Amazon Product Review	18 (S)	Works pretty good.	(A) [PAD] (B) [PAD] (C) [PAD] (D) [PAD] (E) [PAD] (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) [PAD] (L) [PAD] (M) [PAD] (N) [PAD] (O) [PAD] (P) Great value for a creeper. (Q) [PAD] (R) [PAD] (S) Works pretty good. (T) [PAD] [SEP] Just wish the fm stations on the device would go lower. The best one in my area is 85.1 but the device only goes to 88.1. Still a great product.

TABLE 3.1: Examples generated for SSTuning with English Wikipedia and Amazon product review dataset.

Dataset	# Class	# Train	# Val	# Test
Yahoo.	10	1.4M	0	60k
AG News	4	120k	0	7.6k
DBPedia	14	560k	0	70k
20 News.	20	11,314	0	7532
SST-2	2	67,349	872	0
IMDB	2	25k	0	25k
Yelp	2	560k	0	38k
MR	2	8,530	1,066	1,066
Amazon	2	3.6M	0	400k
SST-5	5	8,544	1,101	2,210

TABLE 3.2: Dataset statistics for evaluation datasets.

3.3.2 Evaluation Datasets

We evaluate the models on 4 topic classification (TC) tasks, including Yahoo Topics (yah) [83], AG News (agn) [83], DBPedia (dbp) [83] and 20newsgroup (20n) [84], and 6 sentiment analysis (SA) tasks, including SST-2 (sst2) [85], IMDb (imd) [86], Yelp (ylp) [83], MR (mr) [87] and Amazon (amz) [83], which are binary classification tasks, and SST-5 (sst5) [85], a fine-grained SA task. Detailed data statistics for each testing dataset are presented in Table 3.2.

Following the baselines [2, 19, 88], we report the accuracy on the test set when available, falling back to the original validation set for SST-2. We summarize the dataset statistics for the evaluation datasets in Table 3.2. We download all the datasets from Huggingface [89], except 20newsgroup. For Yahoo Topics, we concatenate the question and answer as inputs. For DBPedia and Amazon, we concatenate the title and content. For 20newsgroup, we follow the recommendations to remove headers, footers, and quotas[3]. However, if the text becomes empty after removing the components, we will use the original text instead.

The verbalizers for each dataset are shown in Table 3.4. We try to unify the verbalizer design for similar tasks. For topic classification tasks, we use the template *"This text is about [].”* after converting the class names to meaningful words. For binary classifications, we use *"It's terrible."* for negative class and *"It's great."* for positive class. For SST-5, we refer to [90] to design the verbalizers. Some of the reformulated text for the evaluation datasets are shown in Table 3.3.

[3] `https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html`

Dataset	**Label**	**Positive Option**	**Reformulated Text**
AG News	3 (D)	This text is about technology.	(A) This text is about politics. (B) This text is about sports. (C) This text is about business. (D) This text is about technology. (E) [PAD] (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) [PAD] (L) [PAD] (M) [PAD] (N) [PAD] (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) [PAD] (T) [PAD] [SEP] REVIEW: 'Half-Life 2' a Tech Masterpiece (AP) AP - It's been six years since Valve Corp. perfected the first-person shooter with "Half-Life." Video games have come a long way since, with better graphics and more options than ever. Still, relatively few games have mustered this one's memorable characters and original science fiction story.
DBPedia	9 (J)	This text is about animal.	(A) This text is about company. (B) This text is about educational institution. (C) This text is about artist. (D) This text is about athlete. (E) This text is about office holder. (F) This text is about mean of transportation. (G) This text is about building. (H) This text is about natural place. (I) This text is about village. (J) This text is about animal. (K) This text is about plant. (L) This text is about album. (M) This text is about film. (N) This text is about written work. (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) [PAD] (T) [PAD] [SEP] Periscepsia handlirschi. Periscepsia handlirschi is a species of fly in the family Tachinidae.
SST-2	1 (B)	It's great.	(A) It's terrible. (B) It's great. (C) [PAD] (D) [PAD] (E) [PAD] (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) [PAD] (L) [PAD] (M) [PAD] (N) [PAD] (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) [PAD] (T) [PAD] [SEP] charles ' entertaining film chronicles seinfeld 's return to stand-up comedy after the wrap of his legendary sitcom , alongside wannabe comic adams ' attempts to get his shot at the big time .
SST-5	3 (D)	It's good.	(A) It's terrible. (B) It's bad. (C) It's okay. (D) It's good. (E) It's great. (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) [PAD] (L) [PAD] (M) [PAD] (N) [PAD] (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) [PAD] (T) [PAD] [SEP] u.s. audiences may find -lrb- attal and gainsbourg 's -rrb- unfamiliar personas give the film an intimate and quaint reality that is a little closer to human nature than what hollywood typically concocts .

TABLE 3.3: Examples after reformulation for 4 evaluation datasets.

3.3.3 Baselines

We choose the following baselines for comparison:

- **Textual entailment (TE) [42]** Following [88], we download the off-the-shelf models trained on MNLI and use the default hypothesis template *"This example is []."* for evaluation.
- **TE-Wiki [43]:** This model is also trained with entailment methods but with a dataset constructed from Wikipedia.
- **Prompting-based method [38]:** We compare with the results using multiple verbalizers reported in [2].
- **Mining-based [2]:** The method has three steps, which are *mine*, *filter* and *fine-tune*. We compare with the results reported.
- **UniMC [19]:** We download the released checkpoint and test the model without question prompts since the reported results on text classification tasks are better on average.

We followed the setups and verbalizers of the original works as much as possible. If the original work does not have verbalizers for a dataset, we will use the same or comparable verbalizers as ours, as shown in Table 3.4.

3.3.4 Implementation Details

To test the performance of the proposed method on different model sizes and architectures, we tune three versions of models, which are based on $\text{RoBERTa}_{\text{base}}$, $\text{RoBERTa}_{\text{large}}$ [46], and $\text{ALBERT}_{\text{xxlarge}}$ (V2) [21], denoted as SSTuning-base, SSTuning-large, SSTuning-ALBERT, respectively. We set the maximum token length as 512 and only run one epoch. We repeat all the experiments 5 times with different seeds by default. The experiments on SSTuning-base and SSTuning-large are run on 8 NVIDIA V100 GPUs and the experiments on SSTuning-ALBERT are run on 4 NVIDIA A100 GPUs.

We set the batch size based on the constraint of the hardware and do a simple hyperparameter search for the learning rate. We do not add hard negatives for

Dataset	**Verbalizers**
Yahoo Topics	"This text is about society & culture.", "This text is about science & mathematics.", "This text is about health.", "This text is about education & reference.", "This text is about computers & internet.", "This text is about sports.", "This text is about business & finance.", "This text is about entertainment & music.", "This text is about family & relationships.", "This text is about politics & government."
AG News	"This text is about politics.", "This text is about sports.", "This text is about business.", "This text is about technology."
DBPedia	"This text is about company.", "This text is about educational institution.", "This text is about artist.", "This text is about athlete.", "This text is about office holder.", "This text is about mean of transportation.", "This text is about building.", "This text is about natural place.", "This text is about village.", "This text is about animal.", "This text is about plant.", "This text is about album.", "This text is about film.", "This text is about written work."
20 Newsgroup	"This text is about atheism.", "This text is about computer graphics.", "This text is about microsoft windows.", "This text is about pc hardware.", "This text is about mac hardware.", "This text is about windows x.", "This text is about for sale.", "This text is about cars.", "This text is about motorcycles.", "This text is about baseball.", "This text is about hockey.", "This text is about cryptography.", "This text is about electronics.", "This text is about medicine.", "This text is about space.", "This text is about christianity.", "This text is about guns.", "This text is about middle east.", "This text is about politics.", "This text is about religion."
SST-2, IMDB, Yelp, MR, Amazon	"It's terrible.", "It's great."
SST-5	"It's terrible.", "It's bad.", "It's okay.", "It's good.", "It's great."

TABLE 3.4: Verbalizers for the evaluation datasets.

the Amazon review dataset since the reviews are not in the format of articles. We also tried to use the negative options from the same product category as hard negatives but did not find any meaningful improvement. We set N_{model} for as 20 and N_{maxLabel} as 10 after simple experiment. The hyperparameters for the main results (Section 4.3.3) are shown in Table 3.5. We try to use the same settings as much as possible. The training time for the three SSTuning models is with 5.12M tuning samples and 64k validation samples (also generated via FSP).

Parameter	Fine-tuning	SSTuning-base/SSTuning-large	SSTuning-ALBERT
Model	$\text{RoBERTa}_{\text{base}}$ (123M)	$\text{RoBERTa}_{\text{base}}$/$\text{RoBERTa}_{\text{large}}$ (355M)	$\text{ALBERT}_{\text{xxlarge}}$(V2)(235M)
Model Selection	Best	Best	Best
Batch Size	16	128	64
Precision	FP16	FP16	FP16
Optimiser	AdamW	AdamW	AdamW
Learning Rate	1e-5	2e-5	1e-5
LR Scheduler	linear decay	linear decay	linear decay
AdamW Epsilon	1e-8	1e-8	1e-8
AdamW β_1	0.9	0.9	0.9
AdamW β_1	0.999	0.999	0.999
Weight Decay	0.01	0.01	0.01
Classifier Dropout	0.1	0.1	0.1
Attention Dropout	0.1	0.1	0
Hidden Dropout	0.1	0.1	0
Max Steps	-	40000	80000
Max Epochs	3	1	1
Hardware	1 NVIDIA V100	8 NVIDIA V100	4 NVIDIA A100
Training time	-	3h/8h	31h

TABLE 3.5: Hyperparameters and training information for full-shot fine-tuning, SSTuing-base, SSTuning-large and SSTuing-ALBERT.

3.4 Results and Analysis

3.4.1 Main Results

	Backbone	L	Topic Classification				Sentiment Analysis						Avg
			yah	agn	dbp	20n	sst2	imd	ylp	mr	amz	sst5	
Fine-tuning❖	$\text{RoBERTa}_{\text{large}}$	-	77.1	95.5	99.2	75.3	95.9	96.4	98.3	91.3	97.2	59.9	88.6
TE-Wiki	$\text{BERT}_{\text{base}}$	✓	56.5	79.4	90.4	53.9	57.3	62.0	58.5	56.2	55.8	24.5	59.5
TE-MNLI	$\text{RoBERTa}_{\text{large}}$	✓	28.6	77.6	60.4	40.2	89.6	90.2	92.8	82.8	92.0	**48.8**	70.3
TE-MNLI	$\text{BART}_{\text{large}}$	✓	48.2	74.8	57.1	35.4	89.0	91.1	93.1	81.4	91.9	47.7	71.0
Prompting*	$\text{RoBERTa}_{\text{base}}$	-	34.1	54.6	51.1	-	81.9	81.8	83.1	78.3	83.5	-	-
Mining-based*	$\text{RoBERTa}_{\text{base}}$	✗	56.1	79.2	80.4	-	85.6	86.7	92.0	80.5	92.0	-	-
UniMC*	$\text{ALBERT}_{\text{xxlarge}}$	✓	-	81.3	88.9	-	**91.6**	**94.8**	-	-	-	-	-
UniMC (Rerun)	$\text{ALBERT}_{\text{xxlarge}}$	✓	59.0	84.3	89.2	43.7	90.1	93.6	94.3	87.3	93	45.6	78.0
SSTuning-base	$\text{RoBERTa}_{\text{base}}$	✗	59.1	79.9	82.7	47.2	86.4	88.2	92.9	83.8	94.0	45.0	75.9
SSTuning-large	$\text{RoBERTa}_{\text{large}}$	✗	62.4	83.7	85.6	56.7	90.1	93.0	95.2	87.4	95.2	46.9	79.6
SSTuning-ALBERT	$\text{ALBERT}_{\text{xxlarge}}$	✗	**63.5**	**85.5**	**92.4**	**62.0**	90.8	93.4	**95.8**	**89.5**	**95.6**	45.2	**81.4**

TABLE 3.6: Main results for 4 topic classification tasks and 6 sentiment analysis tasks. ❖: the original training sets (see dataset sizes in Table 3.2) are used to provide results under supervised settings, served as upper bound, otherwise zero-shot results are reported. *: results are taken from corresponding papers. "Labeled" indicates whether the model uses labeled (✓) or unlabeled (✗) data. "Avg" is the arithmetic mean accuracy of all the datasets. For SSTuning models, we report mean accuracy of 5 repetitions using different seeds.

The main results are shown in Table 3.6. We have the following observations: 1) Our method SSTuning-ALBERT achieves new state-of-the-art results on 7 out of 10 datasets, and even approaches the performance with the supervised setting (i.e.,

	TC	**SA**	**All**
Amazon	63.4	81.4	74.2
Wikipedia	63.4	77.9	72.1
Amazon + Wikipedia	**67.2**	**81.7**	**75.9**

TABLE 3.7: Zero-shot results with different tuning datasets. The best result is in **Bold**.

results of fine-tuning), showing the superiority of our proposed method. 2) With the same backbone, SSTuning-ALBERT outperforms UniMC by 3.4% on average. Note that different from UniMC, we do not utilize any labeled data to conduct meta-tuning, but purely rely on auto-constructed data for self-supervised tuning, which not only has a much large scale of data but also has more abundant options (first sentences). 3) Comparing methods based on $\text{RoBERTa}_{\text{base}}$, $\text{RoBERTa}_{\text{large}}$ and $\text{BART}_{\text{large}}$, our SSTuning-large and SSTuning-base are the two best-performing models on average. We also observe that SSTuning-large outperforms UniMC, which has a stronger backbone. 4) Our models don't perform very well on SST-5, which is a fine-grained sentiment analysis task. Maybe we can generate more fine-grained options from the unlabeled corpus to improve performance on such tasks. We leave it as a future work.

3.4.2 Ablation Study

3.4.2.1 Ablation on Tuning Datasets

We utilize both the Amazon review dataset and English Wikipedia during the tuning stage. To evaluate their effectiveness, we conduct ablation studies to create two model variants that are only trained on one dataset. We set the number of samples for each case to 5.12M for a fair comparison. As shown in Table 3.7, both datasets contribute to the final performance, thus discarding any one leads to a performance drop. It is interesting that tuning with Amazon review data performs the same as tuning with Wikipedia on topic classification tasks. This is unexpected since Wikipedia is more related to topic classification tasks intuitively. We anticipate the reason is that the backbone models have already been pre-trained with Wikipedia, thus further tuning with it does not bring significant advantages.

	TC	SA	All
First sentence prediction	**67.2**	81.7	**75.9**
Last sentence prediction	59.8	**82.2**	73.3
Next sentence selection	54.8	81.9	71.1
Random sentence prediction	56.8	80.8	71.2

TABLE 3.8: Zero-shot results with different tuning objectives. The best results are in **Bold**.

3.4.2.2 Alternative Tuning Objectives

We have proposed first sentence prediction (FSP) as the tuning objective to equip the model learning to associate the label and text in the inference stage. We consider some alternative objectives here to for comparison: 1) last sentence prediction (LSP), which treats the last sentence as the positive option for the rest of the paragraph; 2) next sentence selection (NSS)[4], which treats the first sentence in a consecutive sentence pair as text and the next as the positive option; 3) random sentence prediction (RSP), which randomly pick a sentence in a paragraph as the positive option and treat the rest as text. The comparison between the four settings is shown in Table 3.8. We find that FSP performs the best, especially for topic classification tasks. Among the alternatives, utilizing LSP as the tuning objective leads to the best performance, which is expected since the last sentence in a paragraph usually also contains the central idea, sharing a similar function as the first sentence. Unlike topic classification tasks, the four settings perform similarly on sentiment analysis tasks. The possible reason is that each sentence in a paragraph shares the same sentiment.

3.4.3 Analysis

3.4.3.1 Classification Mechanism

To investigate how our models make correct decisions, we did a case study on a movie review example. As shown in Figure 3.3, we used SSTuning-base (number of labels configured as 2) to classify whether the movie review "A wonderful movie!" is negative or positive. We set the verbalizers as "Bad." and "It's good." to see

[4]Note that we use NSS here to distinguish from NSP (next sentence prediction) used by Devlin et al. [4].

how the length of options impacts the decision. The prediction of the model is 1, which is correct. We focus on a few important tokens, including the classification token `<s>`, the option indicators `A` and `B`, and the separator token `</s>`.

In Layer 0, `<s>` attends to all the options and the text. `A` and `B` attend more to its own options. `</s>` attend more to the text tokens. In higher layers, `A` and `B` attend even more to their own option tokens (Layer 1) but also have some interactions (Layer 4). In layer 9, `A` and `B` attend more its own option tokens again and also the period mark, while `</s>` attend to both the text tokens and the options tokens for `B` (the positive option). In the end, we find that `<s>` token attends more to the second opinion, especially to the tokens around the index indicator "B" in the last layer. This is consistent with our intuitions. For humans, when we do classification tasks, we normally compare the options and select the option that best matches the text. Based on the observations, we hypothesize that the model has the capability to encode the options and text separately, compare the options and text, and choose the positive option in the end.

3.4.3.2 Importance of Index Indicators

To further understand how the index indicator guides the model to make the prediction, we employ different indicator designs during the tuning and inference stage. Specifically, we consider different formats of the index indicator, which are: 1) alphabet characters (A, B, C...), which is the default format; 2) numerical index (0, 1, 2...); 3) same index indicator for all options (0, 0, 0...). During the inference, we also consider two special indicators: 4) same alphabet characters (A, A, A...), and 5) rearranged alphabet characters (B, A, D, C...). The results are shown in Table 3.9. There is not much difference between using alphabet characters and numerical indexes, as shown in cases 1 and 2. As shown in case 3, using the same characters will degrade the performance but not much, which means the model can rely on position embedding of the index indicator to make the correct predictions. As shown in cases 4 and 5, using inconsistent index indicators will greatly degrade the performance, which further verifies the importance of using consistent index indicators to make correct predictions.

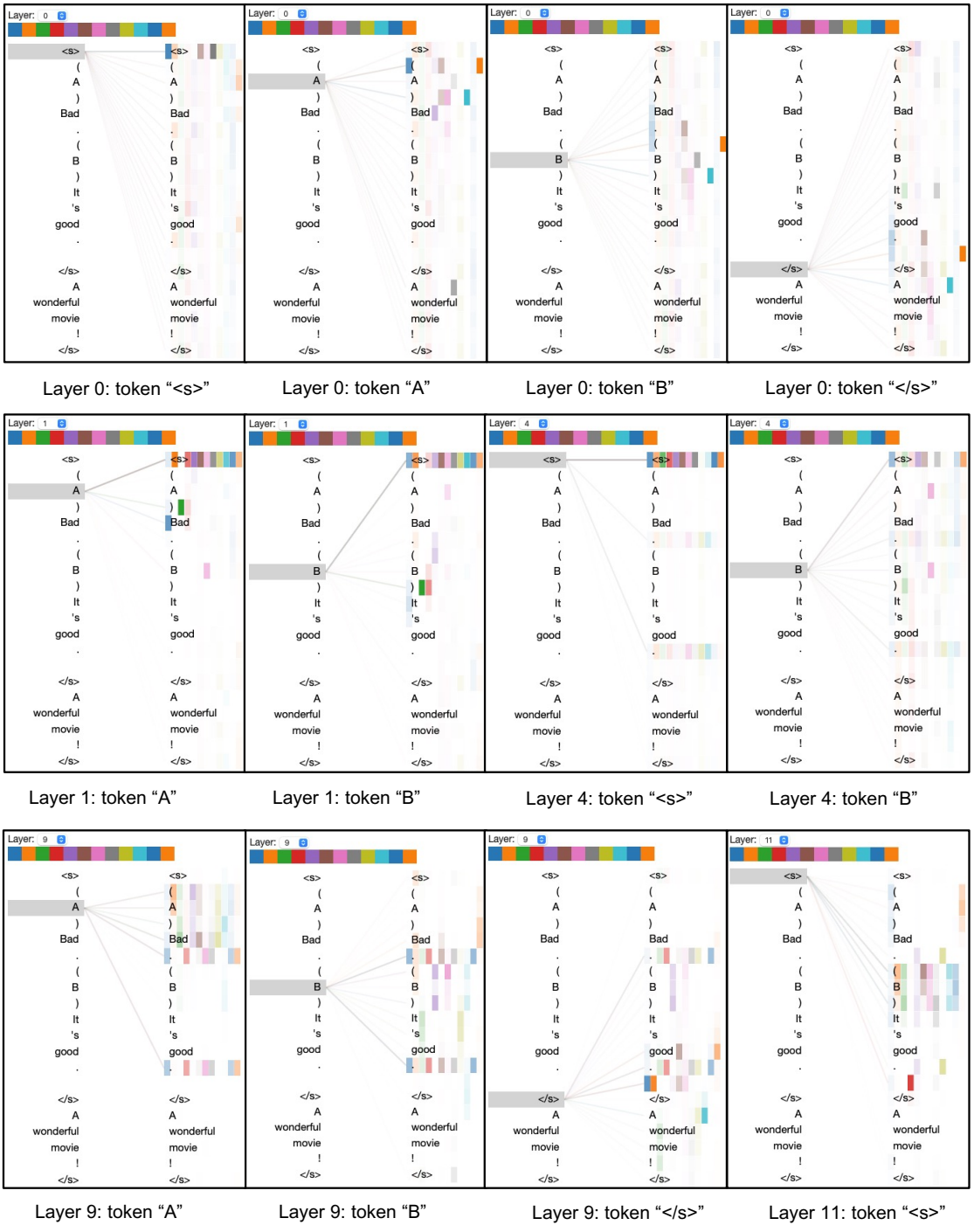


FIGURE 3.3: Attention map for a movie review example. The original text is ”A wonderful movie!” and the verbalizers are ”Bad.” and ”It’s Good.”. The model is SSTuning-base with 2 classes. This figure is generated with BertViz [11].

	Tuning	Inference	Avg	Std
1	(A, B, C...)	(A, B, C...)	75.9	0.3
2	(0, 1, 2...)	(0, 1, 2...)	75.6	0.4
3	(0, 0, 0...)	(0, 0, 0...)	74.1	0.6
4	(A, B, C...)	(A, A, A...)	32.0	1.1
5	(A, B, C...)	(B, A, D, C...)	23.4	12.1

TABLE 3.9: Performance with same and different index indicators during tuning and inference. "Std" indicates Standard Deviation.

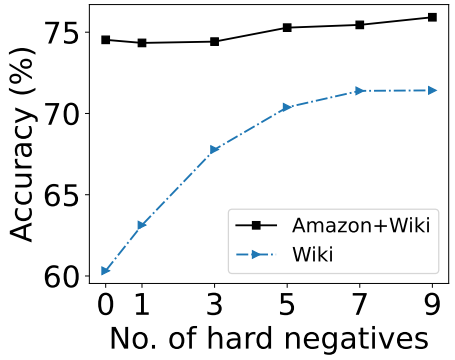


FIGURE 3.4: Zero-shot accuracy with different numbers of hard negatives.

3.4.3.3 Impact of Hard Negative Samples

Intuitively, adding more hard negatives will make the training more difficult, thus forcing the mode to better understand the semantics of the sentences. We tested the impact of hard negatives based on two settings: 1) train with both the Amazon reviews and Wikipedia, each with 2.56M samples; 2) train with only 2.56M Wikipedia samples. We don't train with only Amazon reviews since they don't have hard negatives. The results with $0, 1, 3, 5, 7, 9$ hard negatives are shown in Figure 3.4.

In general, adding more hard negatives will improve the performance. For the case with both datasets, the impact of hard negatives is small. This is because the Amazon review dataset alone can achieve good performance, as shown in Table 3.7. However, hard negatives have a significant impact on the setting with only Wikipedia for tuning. The possible reason is that without hard negatives the model may only learn keyword matching instead of semantics since the keywords may appear many times in the same Wikipedia article.

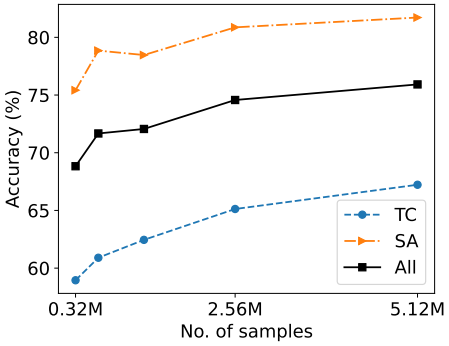


FIGURE 3.5: Zero-shot accuracy with different training sample sizes. Mean accuracy over 4 topic classification tasks, 6 sentiment analysis tasks, and all the tasks are reported.

3.4.3.4 Impact of Tuning Sample Size

To test how the tuning sample size impacts the performance, we trained SSTuning-base with 320k, 640k, 1.28M, 2.56M, and 5.12M samples, with half generated from Wikipedia and half from Amazon reviews. The results are shown in Figure 3.5. With more samples, the performances are increasing in general, especially for topic classification tasks. With such observation, it is likely to further improve the performance by increasing the tuning sample size. Even though tuning on larger datasets is more computationally expensive, it is worth doing since no further training is required for downstream tasks.

3.4.3.5 Impact of Verbalizer designs

During self-supervised tuning, the model saw a large number of first sentences as options, which may contain similar options to the unseen tasks, thus it may have better generalization capabilities. To test how robust the model is to the verbalizer changes compared with UniMC, we design 10 sets of verbalizers for SST-2 and IMDb, covering various scenarios: 1) verbalizers with a single word; 2) verbalizers with different punctuation marks; 3) combinations of single verbalizers; 4) different format for different classes. For a fair comparison, we only use one of our checkpoints and compare it with the UniMC checkpoint released. The results are shown in Table 3.10. We find that SSTuning-ALBERT performs better on average and is more stable. For the most challenging case, which is *"Terrible!"* and *"I like the movie! It is wonderful!"*, SSTuning-ALBERT outperforms UniMC by 20.4 points for SST-2 and 17 points for IMDb.

negative	**positive**	**UniMC(w/o Qn)**		**SSTuning-ALBERT**	
		SST-2	**IMDb**	**SST-2**	**IMDb**
Bad.	Good.	87.0	91.9	90.7	93.9
Terrible.	Great.	88.5	91.7	91.4	94.3
Negative.	Positive.	86.0	90.3	92.2	92.6
Negative!	Positive!	88.9	90.2	92.1	92.4
Terrible!	Awesome!	88.4	91.1	90.9	94.0
Bad, terrible and negative.	Good, great, and positive.	80.7	87.5	87.3	90.8
I don't like the movie!	I like the movie!	91.5	92.9	89.8	90.3
Terrible!	I like the movie! It is wonderful!	66.4	75.1	86.8	92.1
It's terrible.	It's great.	91.6	93.0	90.6	94.1
It's negative.	It's positive.	85.6	89.9	89.2	91.3
	Average	85.5	89.4	**90.1**	**92.6**
	Standard Deviation	7.4	5.3	**1.9**	**1.5**

TABLE 3.10: Comparison of zero-shot results for 2 sentiment analysis tasks with different verbalizers. The best average results are in **bold**.

3.4.3.6 Impact of the Number of Output Labels

In our main results, we set the number of output labels N_{model} as 20. However, a classification dataset may have more than 20 classes. To test the scalability of the label number, we tune another variant for SSTuning-base. We use numerical numbers (0, 1, 2...) as the index indicator and set N_{model} as 40. The comparison between the two versions is shown in Table 3.11. Increasing N_{model} from 20 to 40 only degrade the performance by 1.4 points (75.9% to 74.5%), showing the good scalability of our approach. As an alternative for the datasets with more classes, we can split the labels and do a multi-stage inference.

	N	**Topic Classification**				**Sentiment Analysis**						**Avg**
		yah	**agn**	**dbp**	**20n**	**sst2**	**imd**	**ylp**	**mr**	**amz**	**sst5**	
SSTuning-base	20	59.1	79.9	82.7	47.2	86.4	88.2	92.9	83.8	94.0	45.0	75.9
SSTuning-base	40	58.0	79.3	79.8	49.1	84.4	88.2	91.7	82.2	93.3	39.4	74.5

TABLE 3.11: Accuracy over different number of labels $\boldsymbol{N_{\text{model}}}$ ($\boldsymbol{N}$ means $\boldsymbol{N_{\text{model}}}$).

3.5 Summary

In this chapter, we have proposed a new learning paradigm called SSTuning for zero-shot text classification tasks. By forcing the model to predict the first sentence of a paragraph given the rest, the model learns to associate the text with its label

for text classification tasks. Experimental results show that our proposed method outperforms state-of-the-art baselines on 7 out of 10 tasks. Our work proves that applying self-supervised learning is a promising direction for zero-shot learning.

Limitations

In this chapter, we proposed SSTuning for zero-shot text classification tasks. During inference, we still need to design verbalizers. For simplicity and fair comparison, we only refer to previous works for such designs, which may be sub-optimal. As shown in Table 3.10, using the verbalizers "Terrible." and "Great." work better than "It's terrible." and "It's great." for the SST-2 and IMDA tasks that we reported in the main results. If the labeled validation set is provided, the model may perform better by choosing verbalizers based on the validation set.

Due to limited computation resources, we only tuned the model with 5.12 million samples, which is only a small portion of the available samples. We believe that tuning the model on a larger dataset help improve the performance. Even though the computational cost will also increase, it is worth it since no more training is needed at the inference phase. In addition, we did not do extensive hyperparameter searches except for the learning rate, which may also further improve the performance.

In our experiment, we only tested the method with discriminative models like RoBERTa and ALBERT. Its performance with generative models is not known. It is non-trivial to test on such models since generative models can do both natural language understanding tasks and natural language generation tasks. We leave this as future work.

Chapter 4

Zero-to-Strong Generalization: Eliciting Strong Capabilities of Large Language Models Iteratively without Gold Labels

4.1 Introduction

Pre-trained language models (PLMs) have achieved significant improvements through supervised fine-tuning [91–94]. However, this paradigm often incurs high data costs and requires careful quality control. There are situations where advanced models need to tackle complex tasks that humans cannot fully comprehend or annotate. To study this problem, Burns et al. [8] consider the analogy of using weak models to supervise strong models. By fine-tuning the strong models on the labels generated by the weak supervisors, the strong student model consistently outperforms their weak supervisors, which they call *weak-to-strong generalization*. This phenomenon occurs because strong pre-trained models already possess good representations of relevant tasks.

Despite promising, this *weak-to-strong generalization* paradigm has two limitations. Firstly, the student's performance is still constrained by the supervisor's ability to label the data, and a weaker supervisor leads to a weaker student. Secondly, the reliance on weak supervisor models restricts its applicability to more scenarios. For

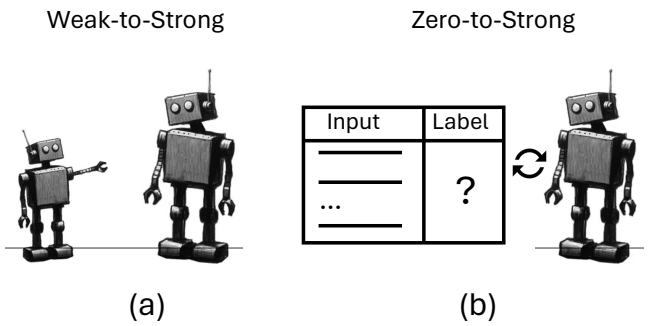


FIGURE 4.1: Illustration of (a) weak-to-strong [8] and (b) our zero-to-strong analogy. While weak-to-strong uses weak models to supervise strong models, zero-to-strong elicits LLM capabilities without ground-truth labels or weak supervisors.

example, there may be cases where no weak supervisors are available or humans cannot provide informative supervision in the future.

To address the aforementioned issue, we explore how to harness the capabilities of LLMs without gold (or ground-truth) labels or weak supervisors, a process we refer to as *zero-to-strong generalization*, as illustrated in Figure 4.1. Previous works have demonstrated that random labels [22, 23] or invalid reasoning paths [24] can also yield good performance, although not as high as with gold labels. Inspired by this, we initially prompt LLMs with random or invalid demonstrations to label the data. We then select a new set of demonstrations based on confidence levels and prompt the LLMs again, repeating this process iteratively. This process allows us to achieve strong performance on tasks without needing gold-labeled data or weak supervisors.

We conducted experiments on 17 classification tasks, 2 extreme-label classification tasks, and 2 reasoning tasks to demonstrate the effectiveness of our proposed methods. Surprisingly, our method not only achieves performance comparable to but even outperforms in-context learning with gold labels for some tasks. We hypothesize that our method selects more suitable samples for demonstrations over iterations, which leads to high performance. Through careful analysis, we find that zero-to-strong learning is more effective for stronger models and more complex tasks. Additionally, it works for fine-tuning and with larger models.

Our main contributions are summarized below:

- We propose a simple yet effective framework called zero-to-strong generalization, which elicits the strong capabilities of LLMs iteratively without gold labels.
- We demonstrate the effectiveness of our zero-to-strong learning with extensive experiments on 17 classification tasks, 2 extreme-label classification tasks, and 2 reasoning tasks.
- We analyze the underlying reasons why zero-to-strong learning is effective and discover that its benefits extend to fine-tuning and larger models.

4.2 Methodology

This section begins with the problem definition, followed by our proposed zero-to-strong learning framework.

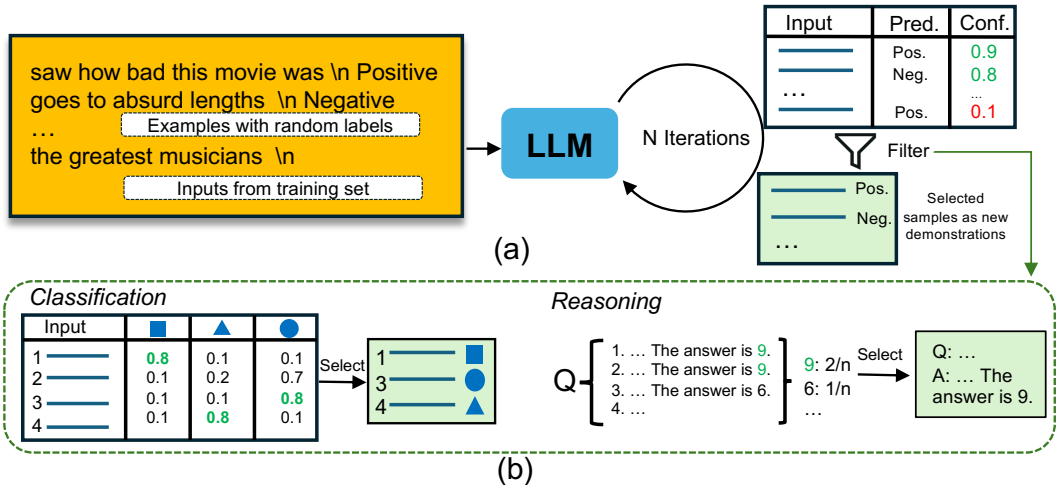


FIGURE 4.2: Illustration of (a) zero-to-strong generalization on a sentiment analysis task and (b) the filtering process. For classification tasks, we select demonstrations by ranking the probabilities for each label. For reasoning tasks, we select the most confident answers based on self-consistency [12].

4.2.1 Problem Definition

In our setting, we assume the absence of gold labels, simulating situations where problems are so complex that human annotations are unreliable. However, we still possess minimal information about the problems. For instance, we know the label space $\mathcal{C}$ in a classification problem, and for a generation problem, the output

format is defined. Additionally, we have access to a few inputs $x_1, \ldots, x_k$ without gold labels.

4.2.2 Zero-to-Strong Generalization

Figure 4.2 illustrates our overall framework, comprising demonstration construction, response generation, sample selection, and iterative evolution.

Demonstration construction. While we lack access to gold labels, we can create demonstrations by randomly sampling from the label space. For classification tasks, labels can be drawn as $\tilde{y} \sim \mathcal{C}$. For reasoning tasks, we can manually generate outputs for a few examples, focusing on maintaining the correct format rather than ensuring complete accuracy.

Response generation. The generated demonstrations are prepended to the input in the training set to form the LLM prompts. By prompting the LLMs, we generate both pseudo labels and their confidence for the training set samples. For classification tasks, we set the temperature to 0 and predict the labels using $\arg\max_{y \in \mathcal{C}} P(y|x)$, where x is the text input and $\mathcal{C}$ is a limited set of potential labels. We use the normalized probability $P(y|x)$ as the confidence. For reasoning tasks, we set the temperature to 0.7 to sample diverse reasoning paths, selecting the most consistent final answer as the prediction. This method is similar to self-consistency [12], and we further calculate the ratio of consistent paths to the total number of paths as the confidence for each sample.

For reasoning tasks, we set the temperature to 0.7 to sample diverse reasoning paths, selecting the most consistent final answer as the prediction. This method is similar to self-consistency [12]. Specifically, given n sampled reasoning paths $\{r_1, r_2, \ldots, r_n\}$ with corresponding answers $\{a_1, a_2, \ldots, a_n\}$, we select the final prediction $\hat{a}$ via majority voting:

$$\hat{a} = \arg\max_{a} \sum_{i=1}^{n} \mathbf{1}[a_i = a] \tag{4.1}$$

We further calculate the confidence score as the ratio of consistent paths to the total number of paths:

$$\text{confidence} = \frac{\sum_{i=1}^{n} \mathbf{1}[a_i = \hat{a}]}{n} \tag{4.2}$$

Sample selection. After generating the responses for all the training samples, we select the k most confident samples for the next iteration. For classification tasks, we uniformly select the top-k most confident samples across the label space to mitigate system bias toward specific classes. For reasoning tasks, we first identify the top-k questions with the highest confidence. Then, for each question, we randomly select one path from the consistent paths. The selection process is illustrated in Figure 4.2(b). Please note that while selected samples may not be perfectly accurate, we observed increased accuracy over iterations, as detailed in Section 4.3.4.2.

Iterative evolution. The selected samples and their predictions will serve as demonstrations for the next round, with this process repeating for several iterations and aiming for progressive performance improvement.

During the evaluation, we set the temperature to 0 and generate final predictions using the same method as in the response generation stage. The zero-to-strong algorithm for classification tasks is detailed in Algorithm 1.

Task	Setting	Llama-3-8B			Mistral-7B		
		4-shot	8-shot	16-shot	4-shot	8-shot	16-shot
Classification	zero-shot	40.7	40.7	40.7	36.1	36.1	36.1
	random label	42.7	50.3	43.8	45.3	51.0	45.9
	gold label	53.3	56.6	61.1	57.5	57.5	**60.5**
	ours (zero-to-strong)	**57.5**	**63.2**	**61.4**	**61.1**	**62.4**	60.1
Extreme-label Classification	zero-shot	21.4	21.4	21.4	**23.9**	23.9	23.9
	random label	4.5	3.7	2.5	5.3	3.6	2.3
	gold label	21.0	26.5	29.1	17.1	**26.1**	26.4
	ours (zero-to-strong)	**24.6**	**27.2**	**33.4**	21.1	23.3	**32.7**

TABLE 4.1: Average Macro-F1 (%) of Llama-3-8B and Mistral-7B on 17 classification and 2 extreme-label classification tasks.

Algorithm 1 Zero-to-Strong

Require: A LLM with $\Pr(y|x)$ accessible.
Require: Input data X, and the label space $\mathcal{C}$
Require: Max iterations M, number of demos K
Initial state: D_0, contains K random labelled demonstrations from X
while Iter $t < M$ **do**
  Calculate $\hat{y} = \arg\max_{y \in \mathcal{C}} P(y|D_{t-1}; x)$;
  Sort the $\hat{Y} = \{\hat{y_1}, ..., \hat{y_i}\}$ in descending order of probability;
  $D_t = \{\}$
  while $|D_t| < K$ **do**
    if $\hat{y_i} \notin D_{t-i}$ **then**
      $D_t = D_t \cup \hat{y}$;
      $i = i + 1$;
    end if
  end while
end while
return $\hat{Y}$

Setting	Llama-3-8B	Mistral-7B
zero-shot	53.5	40.3
invalid	38.9	35.4
gold label	62.2	**53.4**
ours (zero-to-strong)	**64.2**	49.0

TABLE 4.2: Average accuracy (%) of Llama-3-8B and Mistral-7B on reasoning tasks.

4.3 Experiments

We evaluate our proposed framework with two pre-trained LLMs: Meta-Llama-3-8B (Llama-3-8B) [95] and Mistral-7B-v0.1 (Mistral-7B) [96]. All the experiments are conducted on Nvdia A800 GPUs.

4.3.1 Tasks

We assess our framework's effectiveness through three tasks: standard text classification, extreme-label classification, and reasoning. Despite being a subtype of classification, extreme-label classification is treated separately due to its significantly larger class count.

Classification tasks. Following Yoo et al. [23], we evaluate 17 widely-used text classification tasks, with dataset details in Table 4.3. Evaluations are conducted in

4-shot, 8-shot, and 16-shot, using manual templates from Yoo et al. [23]. Examples of the templates are shown in Table 4.6.

Extreme-label classification tasks. Extreme-label classification poses greater challenges than traditional classification due to the large number of labels [97]. For evaluation, we selected the GoEmotions dataset with 28 classes [98] and banking77 with 77 classes [99]. Due to resource limitations, we sampled 1,000 instances from the training set and 500 from the test set. Dataset details can be found in Table 4.4. The templates for the two tasks are shown in Table 4.7.

Reasoning tasks. We choose GSM8k [100] and SVAMP [101] for evaluation, as both require multi-step reasoning. Details of the datasets are in Table 4.5. We selected up to 1,000 samples from the training set and used the entire test set for our experiment. Additionally, we generated 10 diverse reasoning paths for each sample during response generation.

Dataset	#Train	#Test	#C
glue-sst2 [102]	67,349	872	2
glue-rte [103]	2,490	277	2
glue-mrpc [104]	3,668	408	2
glue-wnli [105]	635	71	2
super_glue-cb [106]	250	56	3
trec [107]	5,452	500	5
financial_phrasebank [108]	1,181	453	3
poem_sentiment [109]	843	105	3
medical_questions_pairs [110]	2,438	610	2
sick [111]	4,439	495	3
hate_speech18 [112]	8,562	2,141	4
ethos-national_origin [113]	346	87	2
ethos-race [113]	346	87	2
ethos-religion [113]	346	87	2
tweet_eval-hate [114]	9,000	1,000	2
tweet_eval-stance_atheism [114]	461	52	3
tweet_eval-stance_feminist [114]	597	67	3

TABLE 4.3: Data splits of the 17 classification tasks (#C means number of classes).

Dataset	#Train	#Test	#Classes
GoEmotions [98]	36308	4590	28
banking77 [99]	10003	3080	77

TABLE 4.4: Data splits of the 2 extreme-label classification tasks.

Dataset	# Train	# Test
GSM8k [100]	7473	1319
SVAMP [101]	700	300

TABLE 4.5: Data splits of the 2 reasoning tasks.

Dataset	Manual Template	Verbalizer
glue-sst2	Review: the greatest musicians Sentiment:	negative, positive
glue-wnli	I stuck a pin through a carrot. When I pulled the pin out, it had a hole. The question is: The carrot had a hole. True or False? answer:	True, False
super_glue-cb	That was then, and then's gone. It's now now. I don't mean I 've done a sudden transformation. The question is: she has done a sudden transformation True or False? answer:	True, False, Not sure
trec	Question: What films featured the character Popeye Doyle ? Type:	description, entity, expression, human, number, location
sick	A brown dog is attacking another animal in front of the man in pants The question is: Two dogs are wrestling and hugging True or False? answer:	True, Not sure, False
tweet_eval-hate	Tweet: When cuffin season is finally over Sentiment:	favor, against

TABLE 4.6: Examples of templates for classification tasks. Texts in blue are templates.

Dataset	Template	Verbalizer
GoEmotions	comment: This shirt IS a problem. Get rid of it. emotion category:	admiration, amusement, anger, annoyance...
banking77	service query: When did you send me my new card? intent category:	activate my card, age limit, apple pay or google pay...

TABLE 4.7: Templates for the 2 extreme-label classification tasks. Texts in blue are the templates.

For the 17 classification tasks, we adopt the manual templates and verbalizers from Yoo et al. [23] if possible. Examples for some tasks are shown in Table 4.6. The templates for the two extreme-classification tasks are shown in Table 4.7. The newly created template for "invalid reasoning and answer" is shown in Table 4.11. We keep all the questions the same and modify the reasoning paths and the final answer to make sure they are wrong.

4.3.2 Baseline Methods

We compare zero-to-strong with the following baseline methods:

Zero-shot methods. This setting does not use labeled data as demonstrations. For text and extreme-label classification tasks, predictions are made via $\arg\max_{y \in \mathcal{C}} P(y|x)$, where x is the text input and $\mathcal{C}$ is a limited label set. For reasoning tasks, we adopt the Zero-shot-CoT approach [115], prompting LLMs with "Let's think step by step" and concluding with "Therefore, the answer (Arabic numerals) is" to obtain the final result.

Few-shot with gold labels. For classification and extreme-label classification tasks, we sample k input-label pairs $(x_1, y_1) \ldots (x_k, y_k)$ from the training set either randomly or uniformly based on the label space. We then make predictions via $\arg\max_{y \in \mathcal{C}} P(y|x_1, y_1 \ldots x_k, y_k, x)$. For reasoning tasks, we use a fixed set of demonstrations $(x_1, r_1, y_1) \ldots (x_k, r_k, y_k)$ to prompt LLMs, where r_k represents the reasoning steps, following Wei et al. [15]. The demonstrations are shown in Table 4.8. The final answer is extracted using a regular expression.

Few-shot with invalid labels. In classification and extreme-label classification, demonstrations are generated by assigning random labels rather than using the actual data labels. Each x_i $(1 \leq i \leq k)$ is paired with a randomly sampled label $\tilde{y}_i$ from $\mathcal{C}$. The sequence $(x_1, \tilde{y}_1) \ldots (x_k, \tilde{y}_k)$ is then used to make a prediction by maximizing $\arg\max_{y \in \mathcal{C}} P(y|x_1, \tilde{y}_1 \ldots x_k, \tilde{y}_k, x)$. For reasoning tasks, we reused demonstrations with the "no coherence" setting [24], meaning the rationales are out of order, as shown in Table 4.9.

To ensure reproducibility, we set the evaluation temperature to 0. Results for gold-label, invalid labels, and zero-to-strong are averaged over three seeds to sample the training set for demonstrations. For methods other than zero-shot, initial demonstrations are sampled using two approaches: 1) random initialization — random sampling from the training set, and 2) uniform initialization — sampling an equal number of instances from each class.

4.3.3 Main Results

Table 4.1 presents the main results for classification and extreme-label classification tasks. Our zero-to-strong method for Llama-3-8B consistently outperforms other approaches across all shots settings, demonstrating its effectiveness. It also yields

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: There are originally 3 cars. 2 more cars arrive. 3 + 2 = 5. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: Originally, Leah had 32 chocolates. Her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he gave Denny 20 - 12 = 8. The answer is 8.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. 5 + 4 = 9. The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: There were originally 9 computers. For each of 4 days, 5 more computers were added. So 5 * 4 = 20 computers were added. 9 + 20 is 29. The answer is 29.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: Michael started with 58 golf balls. After losing 23 on tuesday, he had 58 - 23 = 35. After losing 2 more, he had 35 - 2 = 33 golf balls. The answer is 33.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?
A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 5 x 3 = 15 dollars. So she has 23 - 15 dollars left. 23 - 15 is 8. The answer is 8.

TABLE 4.8: Demonstrations for gold label for reasoning tasks [15].

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: Then there were 21 - 15 = 6 trees after the Grove workers planted some more. So there must have been 15 trees that were planted. There are 21 trees originally. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: Then 3 + 2 = 5 more cars arrive. Now 3 cars are in the parking lot. There are originally 2 cars. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: After eating 32 + 42 = 74, they had 32 pieces left in total. Originally, Leah had 74 - 35 = 39 chocolates and her sister had 35. So in total they had 42. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: Then he had 20 - 12 = 8 after giving some to Denny. So he gave Denny 20 lollipops. Jason had 12 lollipops originally. The answer is 8.

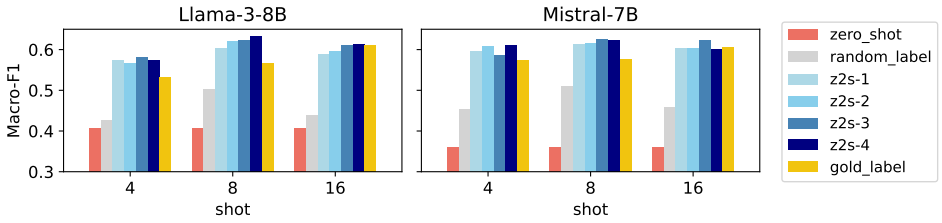
Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Now he has 4 toys. So he got 5 + 4 = 9 more toys. Shawn started with 5 toys. He then got 2 * 2 = 4 toys each from his mom and dad. The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: So 5 computers were added. Now 4 * 5 = 20 computers are now in the server room. There were originally 9 + 20 = 29 computers. For each day from monday to thursday, 9 more computers were installed. The answer is 29.

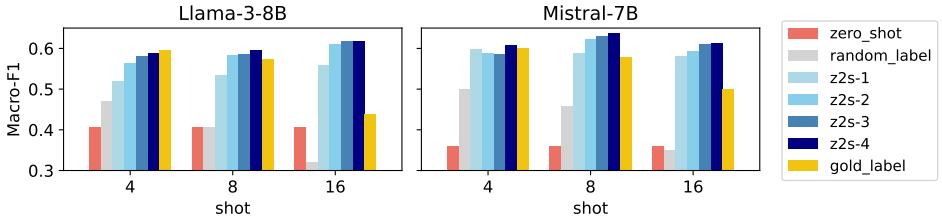
Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: So he had 2 at the end of Tuesday, and 23 at the end of wednesday. He lost 35 - 2 = 33 on Tuesday, and lost 58 more on wednesday. Michael started with 58 - 23 = 35 golf balls. The answer is 33.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?
A: Now she has 5 * 3 = 15 dollars left. So she spent 5 dollars. Olivia had 23 - 15 = 8 dollars. She bought 3 bagels for 23 dollars each. The answer is 8.

TABLE 4.9: Demonstrations for "no coherence" for reasoning tasks.

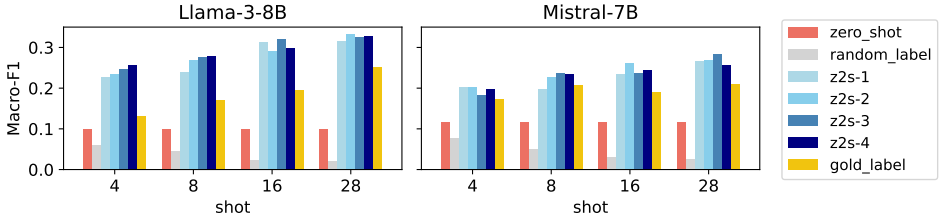


(A) Random initialization.

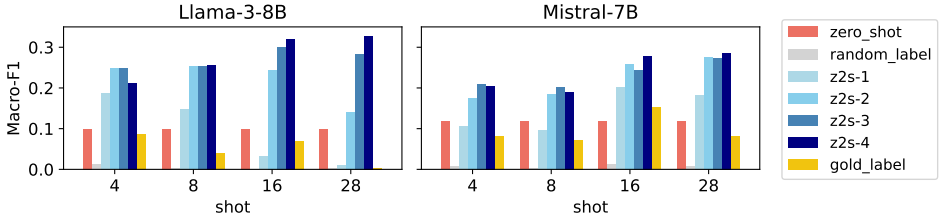


(B) Uniform initialization.

FIGURE 4.3: Average macro-F1 for 17 classification tasks, using two LLMs and two initialization settings. "z2s-i" means the ith round of iteration for zero-to-strong method.



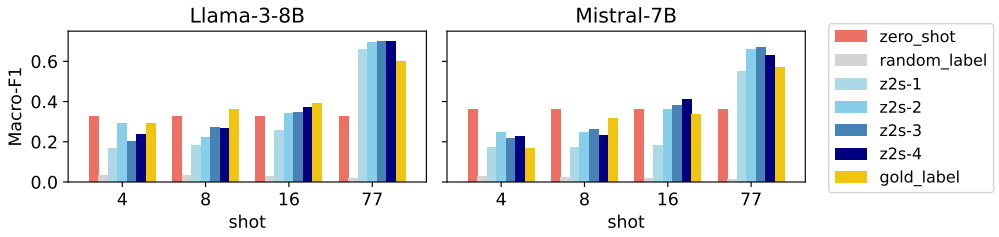
(A) Random initialization.



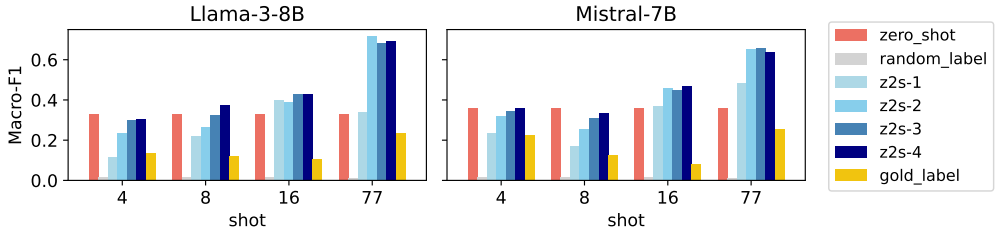
(B) Uniform initialization.

FIGURE 4.4: Average macro-F1 for GoEmotions, using two LLMs and two initialization settings.

the best results with shots lower than 16 for Mistral-7B. We believe this difference stems from Llama-3-8B's superior capabilities, as zero-to-strong performance relies on inherent capabilities gained during pre-training. Overall, extreme-label classification tasks show lower performance compared to standard tasks, emphasizing their increased difficulty. Poor performance in random label settings underlines the necessity of accurate labels for these challenging tasks. Additionally, the number of demonstrations significantly affects extreme-label classification, as performance



(A) Random initialization.



(B) Uniform initialization.

FIGURE 4.5: Average macro-F1 for banking77, using two LLMs and two initialization settings.

with gold-label and zero-to-strong settings improves with more demonstrations, while random-label performance declines.

Table 4.2 presents the average accuracies for the two reasoning tasks. Our zero-to-strong method outperforms other approaches using Llama-3-8B, yet it still lags behind the few-shot method with gold labels using Mistral-7B. This trend aligns with classification and extreme-label classification results, indicating that zero-to-strong is more effective with stronger models. As models continue to improve in the future, our approach may gain even more advantages.

4.3.4 Analysis

The zero-to-strong performance is promising. To better understand its behavior and underlying reasons, we conduct the following analysis.

4.3.4.1 How does the performance improve over the iterations?

Classification tasks. The detailed results for 17 classification tasks are shown in Figure 4.3. It can be seen that for both models, zero-to-strong can achieve comparable or better results than few-shot with gold labels within 4 rounds of

iteration. We hypothesize that the zero-to-strong method selects the most confident samples as demonstrations, which is superior to randomly sampling from gold labels. Zero-to-strong also has a big advantage over few-shot with random labels (please note that few-shot with random labels can be regarded as the 0th round for zero-to-strong). We also notice that for some settings LLMs improve iteration by iteration but the benefits diminish after certain rounds and the performances fluctuate. In addition, the phenomenon exists for all numbers of shots.

Extreme label classification. The results for GoEmotions are shown in Figure 4.4 and the results for banking77 are shown in Figure 4.5. With more demonstrations, few-shots with gold labels perform better with random initialization. It is interesting that when the number of shots is small, few-shot with gold labels underperforms zero-shot setting. We hypothesize that when the number of shots is small, it cannot cover all the labels and make the distribution of the demonstration deviate from the test set. For few-shot with random labels, more demonstrations hurt the performance. This is reasonable as more demonstrations result in more wrong demonstrations, which deteriorate performance. Interestingly, zero-to-strong outperforms few-shot with gold labels in all settings for GoEmotions but the relative performance depends on the initialization settings and the number of shots, which again confirms the effectiveness of zero-to-strong method.

Reasoning tasks. The results for the two reasoning tasks are shown in Figure 4.6. For GSM8K, zero-to-strong improves performance iteration by iteration and approaches few-shot with gold labels after 4 iterations. For SVAMP, zero-to-strong outperforms few-shot with gold labels after a few iterations. We hypothesize that the initial demonstrations with gold label are not optimal for SVAMP and we can generate better demonstrations for this task with zero-to-strong approach.

4.3.4.2 What happens during the iterations?

To further understand the mechanics behind zero-to-strong approach, we conduct more analysis on GoEmotions and GSM8K.

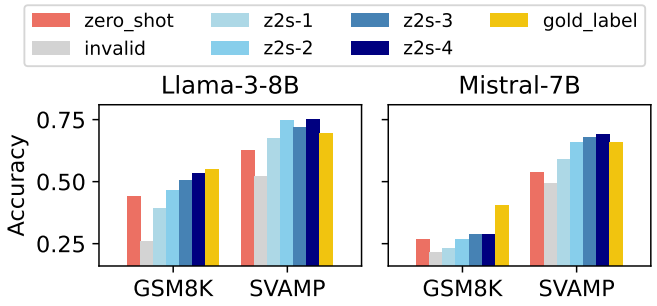
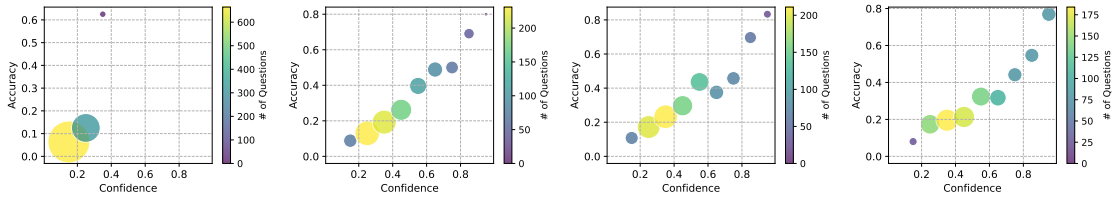
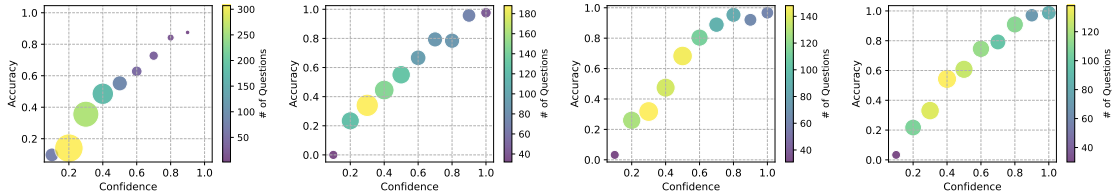


FIGURE 4.6: Accuracy for the two reasoning tasks.



(A) Accuracy vs confidence for GoEmotions from iteration 1 (left) to iteration 4 (right).



(B) Accuracy vs confidence for GSM8K from iteration 1 (left) to iteration 4 (right).

FIGURE 4.7: The relation between accuracy and confidence of the answers for the training set from iteration 1 to iteration 4. The confidence of GoEmotions and GSM8K is calculated based on the methods described in Section 4.2.2. After each iteration, more samples are becoming more confident and accurate.

Does the confidence correlate with the accuracy? Our sample selection process is based on the hypothesis that predictions with higher confidence will have higher accuracy. To verify this hypothesis, we plot the distributions of the sample confidence and their accuracy in Figure 4.7. It can be seen that accuracy is highly correlated with confidence. Initially, more samples have low confidence and low accuracy. After several iterations, more samples have higher confidence and higher accuracy. This observation explains why the model performs better and better.

Do more iterations help with the final performance? In Section 4.3.3, we initially set the maximum number of iterations to 4. In some cases, performance consistently improved with each iteration. However, in other cases, performance reached a plateau after a certain number of iterations and subsequently fluctuated.

To further explore the models' performance over a greater number of iterations, we extended the total number of iterations to 9. The results, depicted in Figure 4.8, indicate that performance does not improve beyond a certain point. We hypothesize that once the optimal demonstrations are selected, additional iterations do not contribute to further improvements.

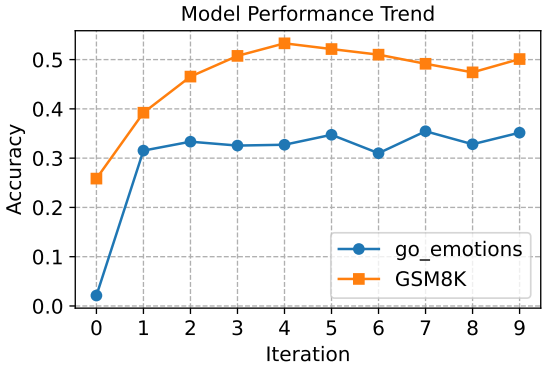
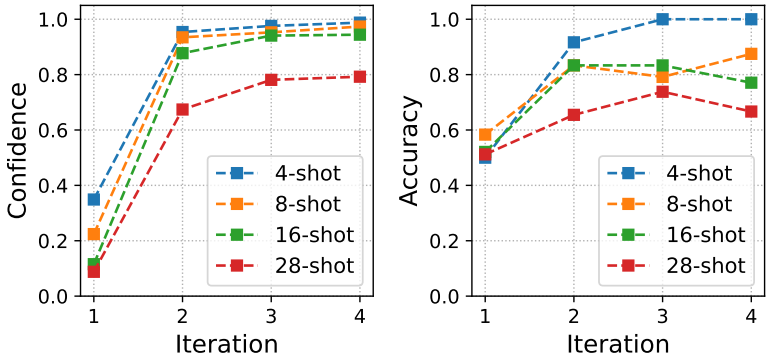


FIGURE 4.8: The accuracy for more iterations for zero-to-strong on GSM8K and GoEmotions. The evaluation is on Llama-3-8B.

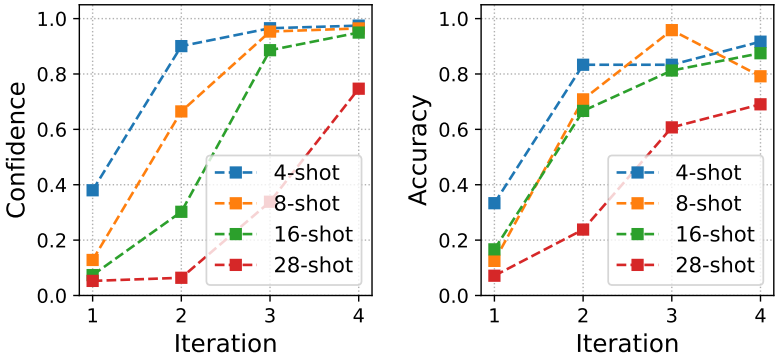
Are the demonstrations more and more confident and accurate over iterations? We select the demonstrations for the next iteration based on confidence. Thus we expect the confidence to increase over iterations. As shown in Figure 4.9, 4.10 and 4.11, the confidence for both GoEmotions and GSM8K increases steadily but saturates after a few iterations. For GoEmotions, confidence for the smaller number of shots is larger and saturates faster. This is expected, as it is harder to get more confident samples. It is also interesting that for GoEmotions, random initialization converges faster than uniform initialization, which is also observed in Figure 4.4. The possible reason is that the training set is not uniform, thus it is better to initialize the demonstrations randomly.

Even though we select the most confident samples for each iteration, we cannot guarantee the accuracy of the selected demonstrations. As shown in Figure 4.9(b) and 4.10(b), the accuracy of the demonstrations fluctuates or even decreases after certain iterations. This is a possible reason why the performances on evaluation sets fluctuate after certain iterations. In certain cases, the same false demonstrations are selected over iterations. We attribute this limitation to inherent constraints in model capability. Nevertheless, despite occasional inaccuracies, these demonstrations still yield meaningful performance improvements.



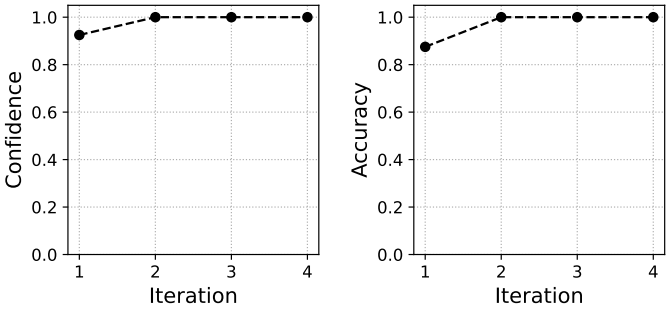
(A) Confidence (B) Accuracy

FIGURE 4.9: Confidence and accuracy of demonstrations over iterations for GoEmotions with random initialization.



(A) Confidence (B) Accuracy

FIGURE 4.10: Confidence and accuracy of demonstrations over iterations for GoEmotions with uniform initialization.



(A) Confidence (B) Accuracy

FIGURE 4.11: Confidence and accuracy of demonstrations over iterations for GSM8K.

Does it work with different initial demonstrations for reasoning tasks? In the previous experiments, we used the "no coherence" demonstration for initialization. To evaluate whether our method applies to general incorrect demonstrations, we tested other settings from Wang et al. [24]. Additionally, we manually

Setting	invalid	z2s-1	z2s-2	z2s-3	z2s-4
Invalid Reasoning	48.8	51.6	54.5	50.7	51.9
No coherence	25.9	46.7	51.0	46.6	51.8
No coherence for BOs	43.9	52.6	54.7	54.2	53.8
No coherence for LTs	29.0	47.3	52.8	52.7	50.3
No relevance	3.9	2.7	2.6	2.7	2.8
No relevance for BOs	37.0	53.2	49.6	51.9	51.6
No relevance for LTs	27.8	47.7	49.4	49.5	51.9
Invalid RnA	38.1	46.0	51.4	51.0	50.9

TABLE 4.10: GSM8K with different invalid demonstrations for Llama-3-8B. The zero_shot score is 44.3, while the few-shot with gold_label is 55.0. "BO" refers to bridging objects and "LT" refers to "language templates". "RnA" refers to "reasoning and answer".

created a new set of demonstrations featuring invalid reasoning and incorrect final answers but containing relevant bridging objects and language templates, as illustrated in Table 4.11. We generate 5 reasoning paths during response generation for this analysis. The results are presented in Table 4.10. From the results, it is evident that the zero-to-strong method achieves accuracies greater than 50% across all settings, except for the "no relevance" condition. This indicates that providing relevant demonstrations is crucial for the zero-to-strong method to be effective. Fortunately, this requirement is manageable for humans, as providing incorrect but relevant reasoning paths and final answers is not hard.

4.3.4.3 Does it work for fine-tuning besides in-context learning?

We further investigate the impact of incorporating fine-tuning with LoRA [116] into our framework. We first generate the labels for the training set with ICL and demonstrations with random labels. Then we filter the samples and fine-tune the model with the pseudo training set. After that, we generate the new labels with the fine-tuned model in a zero-shot manner. We repeat the above process for several iterations. Optionally, we can fine-tune the model with samples labeled after four rounds of zero-to-strong with ICL.

For the fine-tuning experiments, we filter out low-quality training data before each iteration of fine-tuning. For GoEmotions, according to the probability, we retain only the top $\frac{1}{|C|}$ data points for each class from the label space C. This results in duplicated records with different labels. These labels are noisy but still useful for

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 + 15 = 36. The answer is 36.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: There are originally 3 cars. 2 more cars arrive. 3 * 2 = 6. The answer is 6.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: Originally, Leah had 32 chocolates. Her sister had 42. So her sister had 42 - 32 = 10 more chocolates. After eating 35, they had 10 + 35 = 45. The answer is 45.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he has 20 + 12 = 32. The answer is 32.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 2 more toys. 5 + 2 = 7. The answer is 7.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: There were originally 9 computers. 5 more computers were added. So 9 + 5 is 14. The answer is 14.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: Michael started with 58 golf balls. After losing 23 on tuesday, he had 58 - 23 = 35. The answer is 35.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?
A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 3 dollars. So she has 23 - 3 dollars left. 23 - 3 is 20. The answer is 20.

TABLE 4.11: Demonstrations for "invalid reasoning and answer" for reasoning tasks.

tasks	ZS	ft1	ft2	ft3	ft4	z2s-4+ft	GL
GoE	9.9	25.3	26.6	26.7	26.0	31.7	17.2
GSM8K	44.3	30.2	49.7	51.1	50.3	50.3	55.9

TABLE 4.12: Fine-tuning performance for Llama-3-8B. "GoE" refers to "GoEmotions". Results are averaged over 3 seeds. "ZS" refers to "zero-shot". "ft" stands for "fine-tuning". "GL" refers to "gold label".

model	ZS	INV	z2s-1	z2s-2	z2s-3	z2s-4	GL
Llama-3-70B	73.7	30.3	60.7	76.7	80.1	80.7	82.3
Mixtral-8x22B	61.0	19.3	56.7	71.2	72.4	69.8	67.9

TABLE 4.13: Accuracies on GSM8K with larger models. "ZS" refers to "zero-shot". "INV" refers to "invalid". "GL" refers to "gold label".

our fine-tuning process. For GSM8K, we generate 5 paths for each training data and use self-consistency to select confident paths. In all fine-tuning experiments, we set the learning rate to $2e-5$ and train for 3 epochs. As shown in Table 4.12, fine-tuning also improves progressively, notably surpassing few-shot results with gold labels for GoEmotions.

4.3.4.4 Does it work for larger models?

Even though smaller LLMs are more computationally efficient, larger models normally have better performances. To assess the effectiveness of our approach on larger models, we evaluated it on two larger models: Meta-Llama-3-70B (Llama-3-70B) [95] and Mixtral-8x22B-v0.1 (Mixtral-8x22B) [117] on GSM8K. As shown in Table 4.13, zero-to-strong with the two models outperforms the zero-shot setting and achieves comparable performance with few-shot with gold labels, which is consistent with that observed on smaller models, suggesting that our method generalizes well across models of varying sizes.

4.3.5 How does the self-annotation bias impact the model performance?

Figures 4.4 and 4.5 illustrate that varying initialization methods yield performance differences, underscoring the importance of demonstration distribution. As Figures

4.9 and 4.10 indicate, selected samples may contain inaccuracies during iterations. Datasets may exhibit class imbalance if samples are selected solely based on confidence. To address this bias, the sample selection process incorporates class balance considerations, leading to continued performance improvements across iterations. sample selection process. In such setting, the model performance continues to improve over iterations.

4.4 Summary

In this chapter, we propose a new framework called *zero-to-strong generalization*. Without gold label data or weaker supervisors, we can elicit the capabilities of LLMs iteratively through prompting and filtering. Experiments on classification and reasoning tasks demonstrate the effectiveness of this framework. Further analysis shows that by selecting the most confident samples as demonstrations for the next iteration, we also select more accurate and more suitable demonstrations. This framework also generalizes well to fine-tuning and larger models. Our work demonstrates the feasibility of eliciting the capabilities of LLMs with minimal supervision. In the future, we plan to explore *zero-to-strong generalization* in more diverse and challenging tasks.

Limitations

Our framework is restricted to tasks with a single definitive correct answer. For instance, sentences in glue-sst2 [102] can be either positive or negative, and the final answer in GSM8k [118] must be a single number. This uniqueness of the final answer allows us to calculate the confidence of the generated responses. However, for open-ended tasks like story writing, our method is not applicable, as we cannot determine the confidence level of the generated content and leave this as future work. Furthermore, our framework introduces computational overhead during response generation, necessitating hundreds or thousands of inferences prior to sample selection across iterations. Reasoning tasks exacerbate this issue, requiring the generation of multiple reasoning paths to compute confidence scores. However,

this additional cost is specific to the evolution phase; the inference cost remains comparable to few-shot learning employing gold labels.

Chapter 5

Is Translation All You Need? A Study on Solving Multilingual Tasks with Large Language Models

5.1 Introduction

Large language models (LLMs) frequently demonstrate the capability to understand and generate text across multiple languages, a skill attributed to their training on vast corpora composed of texts from various languages [13, 33, 96, 119, 120]. However, these datasets are often disproportionately dominated by English content [9, 25, 26, 121], resulting in an English-centric bias in LLMs. This imbalance can subsequently hinder the models' proficiency in other languages, often leading to suboptimal performance in non-English contexts [27–29].

To enhance performances in multilingual natural language processing (NLP) tasks with English-centric language models, translating training or test data into English has proven an effective strategy [30, 31, 34, 64, 122]. Recent investigations have expanded this idea by incorporating translation, either implicitly or explicitly, into the intermediate stages of prompting LLMs [10, 32, 67] for multilingual NLP tasks. For example, 13 demonstrates that translating test questions into English enhances

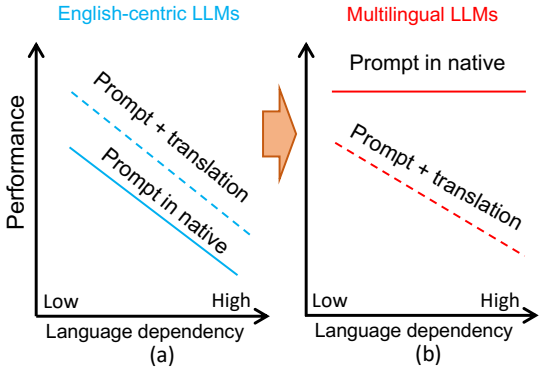


FIGURE 5.1: Illustration of two types of LLMs on tasks with varying language dependencies. "English-centric LLMs" refers to LLMs trained mainly in English corpora. "Multilingual LLMs" refers to ideal LLMs equally capable in all languages.

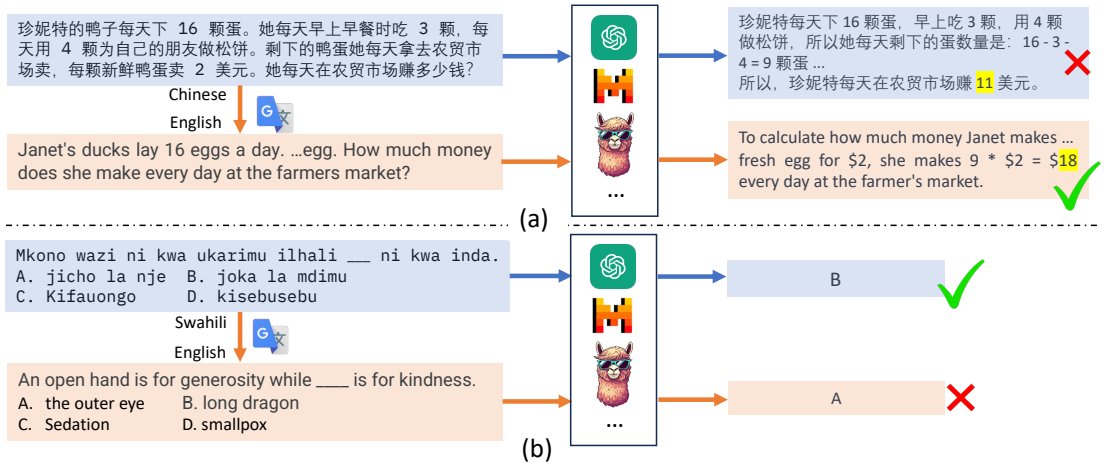


FIGURE 5.2: Examples illustrating how translation can both improve (a) and degrade (b) the performance of LLMs. The Chinese example is from MGSM [13] and the Swahili example is from M3Exam [14]. Translation is beneficial when the questions are semantically equivalent across languages. However, for questions that demand deep cultural knowledge, translation can hinder the ability to answer accurately.

performance on multilingual reasoning tasks, as illustrated in Figure 5.2(a). Similarly, Huang et al. [10] and Etxaniz et al. [67] have shown that prompting LLMs to first translate or comprehend questions in English, then solve them step by step, improves performance.

Despite these advancements, methodologies in various studies differ significantly, and the impact of translation on multilingual task performance remains underexplored. Furthermore, these studies focus on specific NLP tasks and English-centric LLMs, but did not study real-world user queries in various languages. This gap

highlights a need for more nuanced research into the effectiveness of translation techniques across multilingual contexts. As shown in Figure 5.1, we hypothesize that English-centric LLMs generally perform better with English translations of prompts, while "Multilingual LLMs" excel with native prompts, particularly for tasks highly dependent on language.

To address the limitations of existing empirical studies, we perform an in-depth analysis of the utility of translation with large language models for various scenarios. Firstly, we compare translating multilingual tasks into English, with an optional step of translating responses back into the original languages (i.e., the "translate-test" method), against several baselines on multilingual NLP tasks. Secondly, we extend the evaluation to real user queries, which are more likely to contain knowledge related to culture and language. Thirdly, we broaden the scope of LLM evaluations to include non-English-centric models to explore how they differ in behavior from English-centric LLMs. To the best of our knowledge, *this is the first work to analyze the impacts of translating real user queries on multilingual LLMs.*

Our results demonstrate that simply translating queries into English can already achieve the best results in multiple NLP task categories. For real user queries, the effect of translation depends on the languages and the LLMs. When working with advanced LLMs and certain languages, employing prompts in native languages appears to be the more effective strategy. In addition, the non-English-centric LLMs also behave differently from English-centric LLMs, where prompts in the native languages yield superior results by capturing the nuances related to culture and language.

The main contributions of this work are:

- We conduct a comprehensive comparison of multilingual prompting strategies in NLP tasks, finding that translation remains a strong baseline even for LLMs, and identifying factors impacting multilingual performance.
- We expand multilingual evaluation to include actual user queries and and non-English-centric LLMs, addressing the limitations of previous studies.
- We expose critical gaps in current multilingual evaluations, underscoring the need for more comprehensive benchmarks and a broader range of LLMs.

5.2 Translation for NLP Tasks

This section explores various prompting strategies across multiple languages and LLMs, covering a wide range of NLP tasks. This helps us understand how different prompting methods and other factors affect task performance.

5.2.1 Experiment Setup

5.2.1.1 Tasks

We conduct assessments on six benchmarks covering reasoning, understanding, and generation tasks that encapsulate various abilities of LLMs: **MGSM** [13], **XCOPA** [30], **XNLI** [34], **PAWS-X** [123], **MKQA** [124] and **XL-Sum** [125]. Following Huang et al. [10], we choose a subset of 9 languages for MKQA and 5 languages for XL-Sum. For evaluation metrics across our study, we employ the token overlap F1 score specifically for the MKQA dataset, the ROUGE-1 score for assessing XL-Sum, and accuracy as the standard metric for all other benchmarks. Here are the detailed descriptions of the NLP benchmarks:

Arithmetic Reasoning The MGSM [13] benchmark includes mathematical problems from grade school and requires the model to compute the accurate solution. It covers ten languages, and evaluation is conducted using the accuracy score.

Commonsense Reasoning The XCOPA benchmark [30] presents a single premise accompanied by two possible alternatives, and requires selecting which alternative represents the cause or the effect of that premise. It spans 11 languages from diverse language families and is evaluated using an accuracy metric.

Natural Language Inference The XNLI benchmark [34] presents a single premise paired with a single hypothesis; the model must classify the relationship between them as entailment, contradiction, or neutral. It encompasses 15 languages, and performance is assessed using accuracy.

Paraphrase Identification The PAWS-X [123] benchmark presents pairs of sentences and asks models to determine if the pair expresses the same meaning. It spans seven languages, and performance is evaluated using accuracy.

Question Answering The MKQA dataset [124] contains open-domain questions that require predicting short answers. Questions that are unanswerable or excessively long to have a specific answer are not considered during evaluation. This dataset covers 25 languages, with our focus on 9 languages: de, es, fr, ja, ru, th, tr, vi, and zh. We assess the model's performance using the token overlap F1 score.

Summarization The XL-Sum benchmark [125] tasks models with compressing long-form news articles into concise summaries. It encompasses 44 languages; for our experiments, we use a subset of five: es, fr, tr, vi, and zh. Evaluation is performed using the ROUGE-1 metric.

These tasks cover a wide array of 24 diverse languages, including Thai (th), Telugu (te), Greek (el), Arabic (ar), Estonian (et), Urdu (ur), Chinese Simplified (zh), Haitian Creole (ht), Swahili (sw), Bengali (bn), Turkish (tr), Southern Quechua (qu), Russian (ru), Italian (it), Vietnamese (vi), German (de), Japanese (ja), Korean (ko), Bulgarian (bg), French (fr), Indonesian (id), Tamil (ta), Spanish (es), and Hindi (hi). We categorize languages larger than 1% frequency in Common Crawl[1] as high-resource languages (i.e., de, ru, fr, zh, es, ja, it and vi), and the rest as low-resource languages. We exclude English since we want to evaluate the efficient prompting strategy for non-English tasks.

For each task, we sample 500 examples from the test set per language or use the entire test set if there are fewer than 500 examples. For generation tasks like MKQA and XL-Sum, answers will be translated back to the original language if the prompting strategy uses a translator.

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

Model	Prompt type	MGSM		XCOPA		XNLI		PAWS-X		MKQA		XL-Sum		AVG	
		high	low	high	low	high	low	high	low	high	low	high	low	high	low
ChatGPT	Native-Basic	44.4	19.4	84.6	69.7	56.9	48.6	51.6	40.6	35.1	36.4	32.5	29.9	50.8	40.8
	EN-Basic	50.3	27.3	88.3	73.3	64.6	61.8	64.3	50.4	37.4	33.3	**33.3**	**30.0**	56.4	46.0
	Native-CoT	65.1	27.1	84.1	69.8	54.9	47.4	51.6	43.4	35.5	35.1	31.9	27.9	53.8	41.8
	EN-CoT	70.5	47.1	89.9	75.9	60.2	53.6	63.7	51.2	**43.3**	41.2	30.0	28.6	59.6	49.6
	XLT	70.4	50.1	89.3	76.8	60.6	58.1	59.7	58.2	37.7	37.5	22.8	26.1	56.7	51.1
	Trans-Google	**74.7**	**72.7**	**90.3**	**83.2**	**62.4**	**59.1**	68.2	62.0	42.5	**48.3**	30.6	28.9	**61.4**	**59.0**
	Trans-NLLB	65.6	54.1	85.7	78.2	60.5	58.2	**68.4**	**63.4**	35.4	43.6	28.4	27.7	57.3	54.2
Llama-2-70B-Chat	Native-Basic	35.7	5.6	64.2	48.0	43.0	36.0	53.3	50.4	28.9	10.4	30.1	26.8	42.5	29.5
	EN-Basic	42.5	7.7	70.7	52.0	52.7	41.9	61.9	52.8	25.7	21.5	30.2	35.3	47.3	35.2
	Native-CoT	35.5	5.6	65.3	46.8	41.0	35.6	56.0	49.6	25.3	9.9	26.0	25.2	41.5	28.8
	EN-CoT	45.6	7.0	80.7	56.3	52.7	40.9	66.5	57.0	32.7	25.7	29.8	32.0	51.3	36.5
	XLT	49.0	8.4	76.4	54.7	**57.3**	48.4	56.6	51.6	26.5	26.7	19.3	11.5	47.5	33.6
	Trans-Google	**55.5**	**50.0**	**86.3**	**79.7**	55.3	**53.0**	69.4	**64.2**	**38.7**	**43.1**	**33.1**	**36.7**	**56.4**	**54.4**
	Trans-NLLB	46.5	39.7	83.3	75.6	53.7	51.0	**70.5**	62.4	17.8	24.7	32.4	36.2	50.7	48.3

Table 5.1: Average scores of the high-resource languages and low-resource languages for the six benchmarks in zero-shot setting. The best result for each model is in **bold**.

5.2.1.2 Models

We mainly conduct experiments on the following two LLMs, consisting of one closed-source language model and one open-source language model:

- ChatGPT. This is the most capable and cost-effective model in the GPT-3.5[2] family optimized for chat. We chose the latest version (gpt-3.5-turbo-1106) for the experiment.
- Llama-2-70B-Chat. This is the largest chat models in Llama-2 family [126]. Due to computational resource limitations, we use the AWQ [127] version for evaluation.

Besides ChatGPT and Llama-2-70B-Chat, we have also evaluated the NLP tasks with the following models:

- Mistral-7B-Instruct (v0.2). This model is the instructed version of Mistral-7B [96].
- Llama-2-13B-chat, which is a chat model in Llama-2 family [126].
- bloomz-7b1, which is a model fine-tuned with multiple tasks, including some multilingual tasks [119].

[2] `https://platform.openai.com/docs/models/gpt-3-5`

5.2.1.3 Prompting Strategies

We assess experimental strategies based on language of instruction, chain-of-thought reasoning, and translation tools, using a zero-shot approach as the selected models are fine-tuned for instruction-following.

Basic prompt with native instructions (Native-Basic) The questions are posed directly without using prompting strategies like chain-of-thought. Both the query and instructions are presented in their original language.

Basic prompt with English instructions (EN-Basic) Compared with NATIVE-BASIC, EN-BASIC instructs LLMs with English but the query information is in the original language.

Native chain-of-thought (Native-CoT) In NATIVE-COT, we ask the question in the native language and ask the model to reason with the native language with the instruction "*Let's think step by step.*" translated into that language.

English chain-of-thought (EN-CoT) We pose the question in the native language but instruct the model to reason in English with the instruction "*Let's think step by step in English*".

Cross-lingual-thought (XLT) XLT [10] is a state-of-the-art prompting method to handle multilingual NLP tasks. It prompts LLMs to translate the question into English and solve the problem step-by-step in English.

Translate to English with Google Translate (Trans-Google) It uses Google Translate API to translate the original questions into English and then solve the problem step by step.

Translate to English with NLLB models (Trans-NLLB) Instead of using commercial translators, we use an open-source model, namely NLLB [128]. Specifically, we chose `nllb-200-3.3B` to do the translation.

An example of various prompting strategies is shown in Table 5.2. The prompts of EN-Basic for each task are shown in Table 5.3, which are adapted from 10. The prompt templates for other prompting strategies and the instructions for output formats are designed according to the descriptions in Section 5.2.1.3. In addition to the prompting strategies, an output constraint is also included in the template to facilitate answer extraction. When the output format may deviate from the instructions, we utilize "*Therefore, the answer ¡constraint¿ is*" in appropriate languages in the second round to retrieve the ultimate answer.

5.2.2 Main Results

The main results are shown in Table 5.1. We notice that Trans-Google, despite simple, demonstrates the highest overall performance across various models and tasks. While it may not always achieve top performance, it consistently delivers commendable results for both high and low-resource languages. Besides this, we can have the following observations: 1) Utilizing English instructions generally enhances performance across various tasks, regardless of the integration of chain-of-thought. This finding aligns with those reported by 28. 2) chain-of-thought is quite helpful for strong LLMs like ChatGPT and reasoning tasks like MGSM. For weaker models and tasks that can be answered directly, the basic prompt may be a better option. 3) On average, EN-CoT underperforms compared to Trans-Google for both high and low-resource languages. While EN-CoT surpasses Trans-NLLB in high-resource languages, it falls short in low-resource ones. We hypothesize that this discrepancy arises because LLMs excel in high-resource languages but need external translation systems to handle low-resource languages effectively.

These findings are also applicable to smaller models, such as Mistral-7B-Instruct, as demonstrated in Table A.1 in the Appendix. This suggests that the observations generalize well across different model types and sizes. Further results and discussions are provided in Appendix A.1.

Original Question	制作一件袍子需要2 匹蓝色纤维布料和这个数量一半的白色纤维布料。它一共需要用掉多少匹布料
NATIVE-BASIC	{**Original Question** } 您的最终答案的格式应为：”答案: <阿拉伯数字>”.
EN-BASIC	{**Original Question** } You should format your final answer as ”Answer: ¡Arabic numeral¿”.
NATIVE-COT	{**Original Question** } 让我们一步步思考。 您的最终答案的格式应为：”答案: <阿拉伯数字>”.
EN-COT	{**Original Question** } Let’s think step by step in English. You should format your final answer as ”Answer: <Arabic numeral<”.
XLT	I want you to act as an arithmetic reasoning expert for Chinese. Request: {**Original Question**} You should retell the request in English. You should do step-by-step answer to obtain a number answer. You should step-by-step answer the request. You should tell me the answer in this format ’Answer :’.
TRANS-GOOGLE	Crafting a robe requires 2 bolts of blue fiber cloth and half that amount of white fiber cloth. How many pieces of fabric will it use in total? Let’s think step by step. You should format your final answer as ”Answer: <Arabic numeral<”.
TRANS-NLLB	To make a robe, two pieces of blue fiber and half of that amount of white fiber are needed. How many pieces of fabric does it take to make? Let’s think step by step. You should format your final answer as ”Answer: <Arabic numeral<”.

TABLE 5.2: An example of zero-shot prompts for a Chinese problem. For NATIVE-BASIC, EN-BASIC, NATIVE-COT, EN-COT and XLT, we provide the original Chinese question as input and expect an answer in the corresponding format; for TRANS-GOOGLE and TRANS-NLLB, we input the translated question in English, and expect a step-by-step solution in English. To obtain the desirable output format, we instruct the models to output in specific format.

Benchmark	#Test	Basic Prompt
MGSM	250	{problem}
XCOPA	500	Here is a premise: {premise}. What is the {question}? Help me pick the more plausible option: -choice1: {choice1}, -choice2: {choice2}
XNLI	500	{premise} Based on previous passage, is it true that {hypothesis}? 1: Yes, 2: No, or 3: Maybe?
PAWS-X	500	Sentence 1: {sentence1} Sentence 2: {sentence2} Question: Does Sentence 1 paraphrase Sentence 2? 1: Yes, 2: No?
MKQA	500	Answer the question in one or a few words in {target_language}: {question}?
XL-Sum	500	{article} Summarize the article.

TABLE 5.3: Template of EN-BASIC for each benchmark. #Test denotes the number of samples in the test set.

5.2.3 Analysis and Discussions

To investigate the impact of different factors on performance across various languages, we conduct a series of experiments and analyses using the MGSM benchmark.

Is there a relationship between task performance and translation quality? In addition to external translation systems, we can use LLMs to translate the questions. Although XLT includes translation, it is integrated into the solutions. Therefore, we examine the self-translate approach [67], translating in a zero-shot manner with the prompt template:
Translate the following question from {*language*} *to English:*
{*question*}
Don't answer the question, just translate it!
Then we prompt LLMs with the translated question the same as TRANS-GOOGLE and TRANS-NLLB. The results are shown in Table A.3 in the Appendix.

We use the English subset of MGSM as the reference translation and evaluate translation quality using the SacreBLEU score [129, 130]. The results, shown in Figure 5.3, indicate that Google Translate achieves the highest quality for all languages

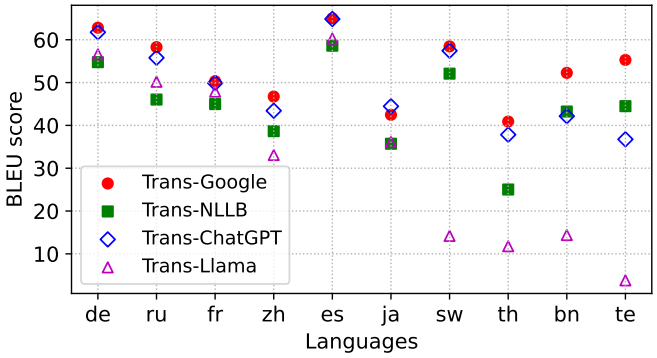


FIGURE 5.3: BLEU scores for translating MGSM questions with different translation systems.

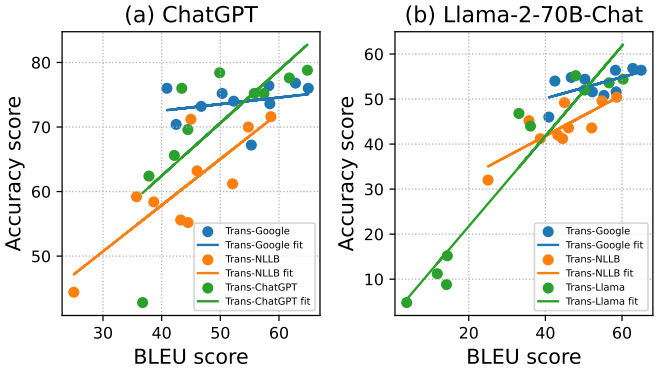


FIGURE 5.4: Corrections between BLEU scores of translation and MGSM accuracy for the three prompting techniques: TRANS-GOOGLE, TRANS-NLLB and self-translate. Each dot in the figure represents the performance of one model on one language.

except Japanese. Translations by ChatGPT (Trans-ChatGPT) and Llama-2-70B-Chat (Trans-Llama) outperform TRANS-NLLB for high-resource languages but not for some low-resource languages.

To analyze the impact of translation quality on final performance, we plot the correlation between accuracy scores and BLEU scores for each language in Figure 5.4. The results show that higher translation quality (BLEU scores) generally leads to better task performance, highlighting the importance of an effective translation system.

Does language distance between English and target language affect the performances? Table 5.1 shows that the LLMs perform better for high-resource languages than low-resource languages on average. We hypothesize that language

Prompt type	SYN	GEO	INV	GEN	PHON
ChatGPT					
Native-Basic	-0.786*	-0.336	0.323	-0.403	-0.044
EN-Basic	-0.820*	-0.160	0.527	-0.299	0.020
Native-CoT	-0.795*	-0.184	0.479	-0.313	0.045
EN-CoT	-0.841*	-0.286	0.339	-0.436	-0.034
XLT	-0.787*	-0.113	0.445	-0.284	0.117
Llama-2-70B-Chat					
Native-Basic	-0.688*	-0.369	0.250	-0.323	-0.044
EN-Basic	-0.782*	-0.512	0.134	-0.513	-0.226
Native-CoT	-0.706*	-0.403	0.231	-0.475	-0.105
EN-CoT	-0.737*	-0.510	0.206	-0.445	-0.219
XLT	-0.697*	-0.432	0.266	-0.423	-0.153

Table 5.4: Pearson correlation coefficient between MGSM accuracy and five language distances between English and that language. A lower value indicates higher correlation due to the negative coefficients.(*$p < 0.05$, two-tailed)

distance, besides language frequency, is crucial for English-centric LLMs. To verify this, we calculate the correlation between MGSM accuracy and the language distances between the target languages and English. Following 131, we examine five types of distances, including the syntactic (SYN), geographic (GEO), inventory (INV), genetic (GEN), and phonological (PHON) distances extracted using `lang2vec` [132]. As shown in Table 5.4, MGSM accuracy significantly correlates with syntactic distance but not with other types of distances. The negative values indicate that languages with a larger syntactic distance from English tend to perform worse.

5.3 Translation for Real User Queries

NLP tasks typically focus on specific linguistic aspects, which may not fully encapsulate the breadth and complexity of real-world user queries which cover diverse topics and require nuanced comprehension. Moreover, these benchmarks are often constructed by translating from the English data [13, 30, 34, 123, 125]. This approach leads to datasets that are not truly challenging, as they miss the rich culture-specific elements crucial for truly nuanced language understanding for different languages. To assess the impact of translation on real-world queries, we

[System]
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, expected language and level of detail of the response. Begin your evaluation by providing a short explanation (up to 100 words). Be as objective as possible. After providing your explanation, please rate the response on a scale of 1 to 10 by strictly following this format: "Rating: <rating>", for example: "Rating: 5".

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]

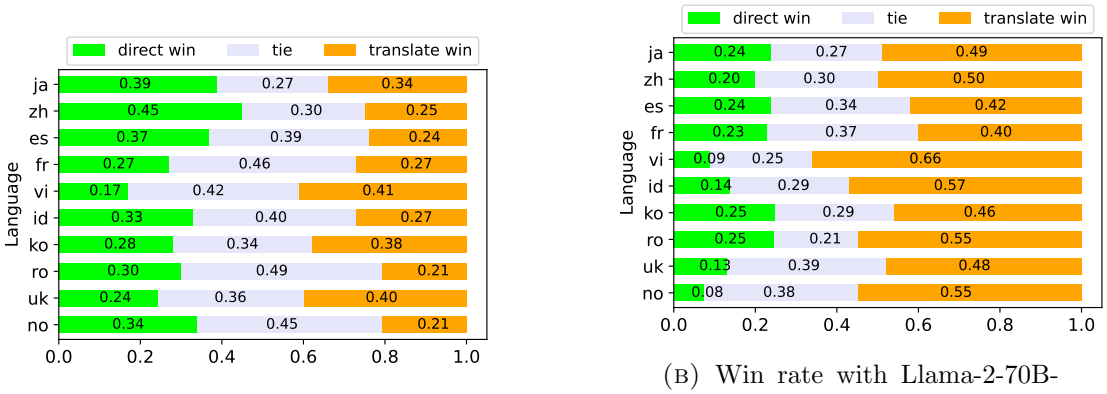
FIGURE 5.5: The LLM-as-a-judge prompt for GPT-4o.

extract user requests from ShareGPT[3], a website to share real conversations with ChatGPT.

5.3.1 Experiment Setup

We selected 10 languages, ranging from high to low resource, and randomly sampled 100 requests for each language. However, for Romanian (ro), Ukrainian (uk), and Norwegian (no), we sampled 53, 98, and 53 requests respectively, due to the limited number of samples available from the source dataset. Since the queries can be in various formats, we only compare two prompting strategies: 1) original queries; and 2) translated queries with Google Translate API. For the second option, we translate the output back to the original language for consistency. To evaluate the quality of the responses, we use GPT-4o[4](`gpt-4o-2024-05-13`) as the judge. The prompt for the judge is shown in Figure 5.5, which is adapted from [36]. With this prompt, each response will get a score from 1 to 10.

[3] `https://sharegpt.com/`
[4] `https://openai.com/index/hello-gpt-4o/`



(A) Win rate with ChatGPT

(B) Win rate with Llama-2-70B-Chat

FIGURE 5.6: Win rate comparison for each language using ChatGPT and Llama-2-70B-Chat.

5.3.2 Main Results

We compared the scores of two response sets from the same model, calculating the win rate for each language. The results are shown in Figure 5.6, leading to the following observations: 1) ChatGPT's performance varies across languages. For high-resource languages like Japanese, Chinese, and Spanish, original queries have a higher win rate. In contrast, for low-resource languages, the effectiveness of translation can be either better or worse, depending on the specific languages involved. 2) For Llama-2-70B-Chat, translation has a higher win rate for all languages, reflecting its English-centric nature. Despite potential information loss, the improved understanding after translation still enhances performance.

Llama-2-70B-Chat and ChatGPT exhibit distinct behaviors, reflecting their inherent differences. Llama-2-70B-Chat, being English-centric, performs better with translated inputs. Conversely, ChatGPT shows certain characteristics of a "Multilingual LLM", as shown in Figure 5.1(b), mainly for high-resource languages, indicating the potential for improvement in true multilingual processing.

To determine if answering user queries requires local cultural knowledge, we used GPT-4o with a specially crafted prompt to analyze queries in multiple languages (Figure 5.7). Results in Table 5.5 show that 30% to 74% of queries per language require cultural knowledge, highlighting the rich cultural elements in the data. Further analysis of the ShareGPT subsets requiring local cultural knowledge is in Appendix A.2. We also conduct additional experiments, detailed in Appendix A.2.1, to verify that advanced LLMs can reliably assess the quality of responses.

Given the following user request, evaluate whether answering this request necessitates local cultural knowledge in that language. Consider factors such as cultural practices, beliefs, historical context, language nuances, and societal norms. Begin your evaluation by providing a short explanation (up to 50 words). After providing your explanation, provide a final assessment of whether cultural knowledge is essential for an accurate and comprehensive answer. Give you final answer in the format "Answer: [Yes/No]".

User Request:
{question}

Question:
1 / 1 将这段富有戏剧性的对话润色成一个剧本中的对话，要求总句数不少于 8 句，总字数不少于 200 字，内容有戏剧冲突和看点：\nA：你在这里做什么？我叫你来帮我搬东西，不是陪我聊天的。\nB：抱歉，我正在回复一个重要的邮件，马上就好。\nA：我又不是没看到你一直盯着手机，你有多少个重要的邮件？\nB：真的很重要，稍等一下，我处理完就马上来帮忙。

GPT-4o response:
Explanation: The request involves transforming a dramatic dialogue into a script, which requires understanding the nuances of the Chinese language, including tone, context, and cultural expressions. Additionally, the request specifies the need for dramatic conflict and appeal, which are influenced by cultural storytelling practices and societal norms.\n\nAnswer: Yes

FIGURE 5.7: Prompt template to check whether answering a request needs local cultural knowledge (upper) and one Chinese example (lower).

Language	**ja**	**zh**	**es**	**fr**	**vi**	**id**	**ko**	**ro**	**uk**	**no**
Ratio (%)	59	58	38	41	67	55	55	74	30	57

TABLE 5.5: The percentage of the questions that necessitate local cultural knowledge.

5.3.3 Analysis and Discussions

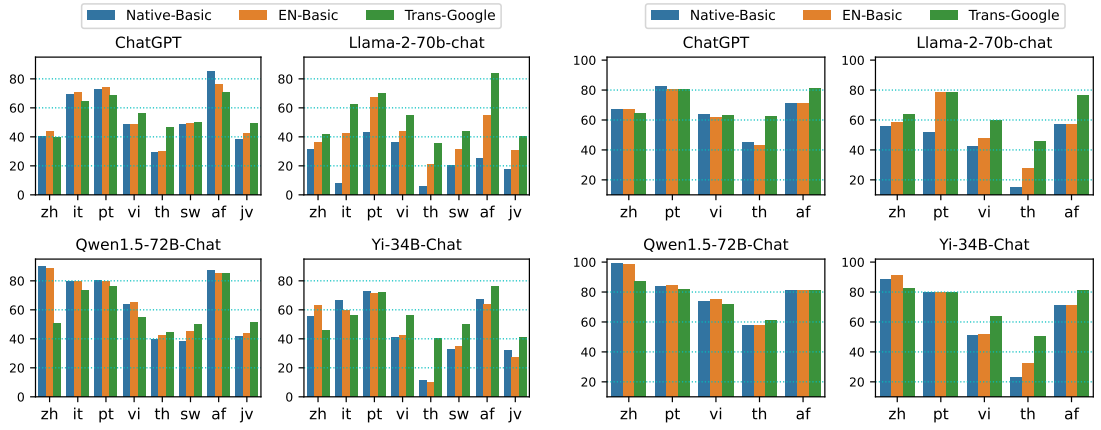
Based on the previous results, ChatGPT and Llama-2-70B-chat both tend to be English-centric but ChatGPT demonstrates certain behaviors of a "Multilingual LLM". Consequently, we broaden our analysis to include non-English-centric LLMs and assess their performance across various tasks.

How do non-English-centric LLMs perform on culture-related tasks? To investigate the behaviors of different LLMs on culture-related tasks, we select another two LLMs: Qwen1.5-72B-Chat [133] and Yi-34B-Chat [134], which are not English-centric. These two open-source models demonstrate strong capabilities in both English and Chinese. Therefore, we can check whether they demonstrate multilingual behaviors in Chinese, as illustrated in Figure 5.1(b).

For the evaluation dataset, we choose M3Exam [14], as the questions are real-world natural data from different languages instead of translating from English

and require strong multilingual proficiency and cultural knowledge to perform well. For example, the question about a Swahili proverb in Figure 5.2(b) requires local knowledge to answer correctly. We select the `language` and `social science` subject categories, which likely contain more native cultural knowledge, and evaluate up to 500 samples per language.

Based on the results shown in Figure 5.8, we have the following observations: 1) For ChatGPT, translation may not always result in improved performance. This observation aligns with the conclusions in the study by Zhang et al. [14]. The effectiveness of translation largely depends on whether translation errors outweigh any potential gains in better comprehension. 2) Translation helps Llama-2-70B-chat in all the languages, suggesting that the model's underperformance is due to poor language understanding rather than limitations of cultural knowledge. 3) Qwen1.5-72B-Chat and Yi-34B-Chat excel in Chinese proficiency. The translation hurts Chinese performance, highlighting the significant influence of translationese on comprehension. Despite this, it may boost performance in other languages, notably for Yi-34B-Chat, indicating that they are far from ideal multilingual LLMs.



(A) (B)

FIGURE 5.8: Accuracies of four LLMs on M3Exam (a) `language` and (b) `social science` subject categories. In M3Exam, not all subjects are available in every language, causing a difference in language coverage between the two subjects.

How do non-English-centric LLMs perform on NLP tasks? As shown in Figure 5.2(b), for an ideal multilingual LLM, prompting in native languages should still have advantages over translation if the tasks are less dependent on languages. To test the hypothesis, we evaluate Qwen1.5-72B-Chat and Yi-34B-Chat on the

Prompt type	Qwen1.5-72B-Chat							Yi-34B-Chat						
	MGSM	XCOPA	XNLI	PAWS-X	MKQA	XL-Sum	AVG	MGSM	XCOPA	XNLI	PAWS-X	MKQA	XL-Sum	AVG
Native-Basic	78.8	93.0	55.8	71.8	36.6	**41.3**	62.9	63.2	92.6	46.0	43.6	13.4	**36.9**	49.3
EN-Basic	77.2	97.0	73.0	**73.0**	32.7	39.7	65.4	66.8	93.6	52.6	**74.6**	15.5	35.1	56.4
Native-CoT	**83.2**	95.8	46.4	72.2	35.8	39.5	62.1	65.2	91.8	42.6	43.6	13.0	36.6	48.8
EN-CoT	81.6	97.2	71.2	70.6	34.9	38.6	**65.7**	70.0	93.6	48.2	74.8	12.1	33.1	55.3
XLT	78.4	**97.8**	**77.4**	67.6	20.8	35.3	62.9	56.0	93.2	**69.2**	65.6	7.5	31.3	53.8
Trans-Google	81.6	94.6	63.8	68.4	**45.7**	31.3	64.2	**71.2**	**94.0**	49.6	70.8	**24.5**	36.3	**57.7**
Trans-NLLB	58.8	88.2	61.4	70.4	32.0	28.5	56.5	56.0	86.6	48.8	68.2	22.9	28.5	51.8

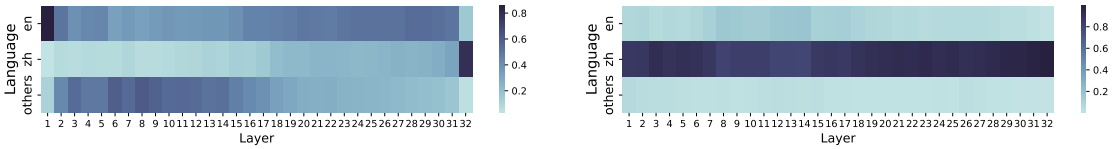
Table 5.6: Scores of the two non-English-centric LLMs on NLP tasks for the Chinese language. The best result for each model is in **bold**.

NLP tasks as discussed in Section 5.2.1.1. We only evaluate them in Chinese since the two models are optimized for this language.

The results are displayed in Table 5.6. Trans-Google remains competitive among various prompting strategies, achieving the best average scores for Yi-34B-Chat, which surpasses our expectations. The possible reason could be that while both models are optimized for Chinese, their performance in Chinese still lags behind their proficiency in English. Nevertheless, We have the following special observations for the two models. 1) For Qwen1.5-72B-Chat, the best strategy is EN-CoT instead of Trans-Google. We hypothesize that this prompting strategy utilizes the model's bilingual abilities and simultaneously avoids translationese. 2) Both LLMs perform better with Native-Basic for the XL-Sum dataset. We hypothesize that the dataset is more language-dependent than other tasks as it is created by considering the local context instead of simply translating from the English version [125]. 3) The translation benefits are less pronounced than those of ChatGPT and Llama-2-70B-Chat. For example, the gap between Trans-Google and Native-Basic on MGSM(Chinese) for the two models are 2.8% and 8%. The values for ChatGPT and Llama-2-70b-Chat are 37.2% and 16%, respectively, which are significantly larger.

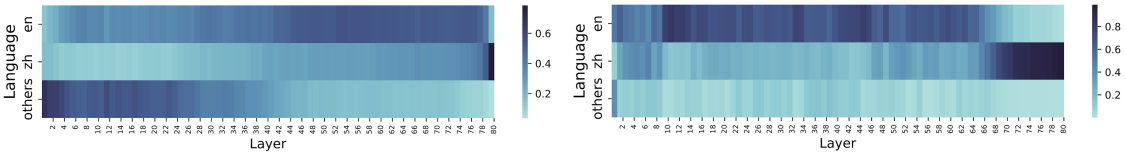
How do different LLMs handle multilingual prompts? To further understand the differences between English-centric LLMs and non-English-centric LLMs, we analyze the layerwise language distribution for Llama-2-7B-Chat and Qwen1.5-7B-Chat, using the method proposed by Zhao et al. [135]. We decode the embedding after each layer and identify each token into different languages with CLD3[5]. As shown in Figure 5.9, the two LLMs process Chinese prompts differently. While

[5] `https://github.com/google/cld3`



(A) Llama-2-7B-Chat (B) Qwen1.5-7B-Chat

FIGURE 5.9: Layerwise language distribution for (a) Llama-2-7b-chat and (b) Qwen1.5-7B-Chat with Chinese prompts.



(A) Llama-2-70B-Chat (B) Qwen1.5-72B-Chat

FIGURE 5.10: Layerwise language distribution for (a) Llama-2-70b-Chat and (b) Qwen1.5-72B-Chat with Chinese prompts.

the hidden representations of Qwen1.5-7B-Chat are mainly in Chinese, those of Llama-2-7B-Chat are in various other languages. We hypothesize that processing the information in native without conversion avoids the information loss, making it more suitable for processing multilingual tasks.

Figure 5.10 illustrates the layerwise language distribution in larger models, including Llama-2-70B-Chat and Qwen1.5-72B-Chat. Llama-2-70B-Chat exhibits the same phenomenon as its smaller counterpart, Llama-2-7B-chat, with diverse languages represented in its hidden states. In contrast to Qwen1.5-7B-Chat, the hidden representations of Qwen1.5-72B-Chat incorporate both Chinese and English until the last few layers, possibly reflecting the challenges of constructing such a large model using Chinese exclusively for hidden representations. Nevertheless, it still represents its hidden states more in Chinese than Llama-2-70B-Chat.

5.4 Summary

We have conducted a thorough evaluation of LLMs in various multilingual tasks. These tasks include traditional NLP benchmarks, real user queries, and culture-related tasks. Even though translation-based methods are simple and effective strategies to overcome the limitations inherent in English-centric LLMs, they are not optimal for all scenarios, highlighting the necessity of more comprehensive

multilingual evaluation. The experiment on non-English-centric LLMs and culture-related tasks demonstrates that employing prompts in the native language emerges as a more effective approach. This method is particularly adept at capturing the subtleties and intricacies unique to each language. The challenge of the setting is that it requires LLMs to be proficient in various languages, calling for the prioritization of research and development efforts toward the creation of strong multilingual LLMs.

Limitations

This study aims to systematically assess the effectiveness of various prompting strategies across different tasks and LLMs. Due to limitations in computing resources, it was not possible to evaluate all existing prompting strategies comprehensively. However, we endeavoured to cover the most commonly employed strategies to formulate a broad conclusion. In our evaluation of LLMs on culture-related tasks, we specifically selected two LLMs optimized for Chinese, acknowledging it as one of the most widely spoken languages globally. The dataset used, M3Exam, comprises exclusively multiple-choice questions. It is important to note this specificity as it may influence the applicability of our findings. In our evaluation, we limited our sampling to up to 500 samples for each language within the benchmarks to manage computational constraints and ensure a broad yet feasible analysis scope. Consequently, our results might not be directly comparable with other studies that evaluate performance across the entire benchmark. In future work, we plan to extend our evaluation to LLMs optimized for other languages and to explore benchmarks presented in various formats beyond multiple-choice questions.

Chapter 6

SeaExam and SeaBench: Benchmarking LLMs with Local Multilingual Questions in Southeast Asia

6.1 Introduction

Large Language Models (LLMs) have shown remarkable performance across various English benchmarks, including both human exam datasets such as MMLU [16], or instruction-following datasets such as MT-Bench [36], indicating their strong capabilities [33, 95, 136]. As these LLMs are increasingly deployed globally, there is growing interest in their ability to handle multiple languages and adapt to a wide range of multilingual applications [10, 120, 137–140].

This led to the development of multiple multilingual benchmarks to assess the multilingual capabilities of LLMs [14, 27, 28]. Among them, many datasets such as MGSM [13], XNLI [34], and Multilingual MMLU [16, 33] are typically constructed by translating the English set into target languages. Considering that original English test sets are often carefully designed, such translations provide an effective way to leverage the task categorization, evaluation targets, and construction methods of the monolingual dataset into the multilingual context.

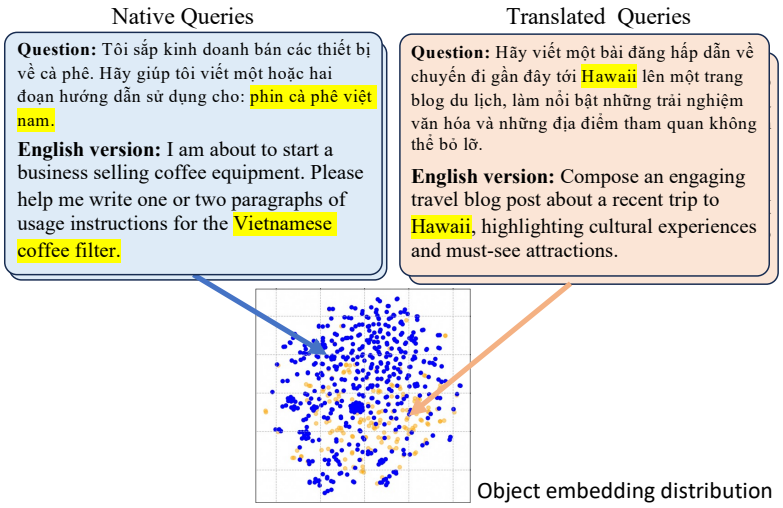


FIGURE 6.1: Compared with local usage queries in Vietnamese, questions in English-based translations show more American context (Hawaii). To better illustrate this discrepancy, we extracted the object in these questions and visualised their distribution. The results show that the objects in translated questions cover only a small portion of those in local usage queries.

However, such translated questions focus merely on evaluating the same contextual elements as their monolingual counterparts. In other words, they focus primarily on the application scenarios relevant to the original benchmarks rather than adapting to a wide range of multilingual applications in the real world. Instead, a truly effective multilingual benchmark must also consider the content typically used in the practical application of the target language [35]. For example, as shown in Figure 6.1, we visualize the distribution of objects in questions collected from local usage queries versus those translated from English. Compared to local usage queries, translated questions based on English exhibit more of an American context, e.g., involving the place "Hawaii". It shows that translated questions cover only a small portion of the entities in local usage queries, indicating a significant divergence in the query context.

Considering the scarcity of such effective multilingual benchmarks, this paper introduces two new benchmarks, SeaExam and SeaBench. These benchmarks are specifically designed to address the unique application scenarios and cultural contexts of Southeast Asian (SEA) countries, which often differ significantly from western-centric datasets. Following the design principles of two widely used English-based datasets, MMLU and MT-bench, we do not simply translate the original English questions but incorporate real-world usage scenarios from SEA natives into the

content — allowing us to measure a model's adaptability in multilingual application scenarios. Specifically, SeaExam is a multitask exam dataset sourced from real exams in SEA countries that cover a wide range of subjects including local history, geography, and literature. SeaBench, following MT-Bench's approach, focuses on multi-turn instruction-following tasks spanning ten task categories. It incorporates scenarios and instructions that are commonly encountered in SEA cultures and daily life.

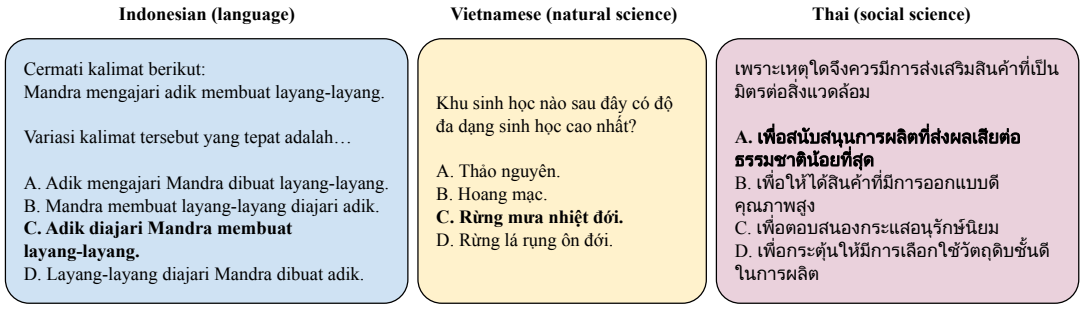
Our experimental analysis quantitatively demonstrates that, **1)** Compared to the translated benchmarks MMLU and MT-bench, our SeaExam and SeaBench benchmarks include questions that are more aligned with the daily usage of regional languages (Section 6.3.1). **2)** Furthermore, using SeaExam and SeaBench, we are able to more effectively discern the capabilities of models in real-world multilingual applications (Section 6.3.2.1). Further analysis reveals that **3)** While multiple-choice questions in exam datasets can objectively measure model capabilities, open-ended questions are more effective in highlighting differences in model performance across various languages (Section 6.3.2.2 and Section 6.3.2.3). Additionally, we find that **4)** The nine models involved generally perform poorly in the "safety" category — evaluating whether the models generate harmful responses in the local context (Section 6.3.2.4). Therefore, we advocate for enhanced safety measures in multilingual applications to adapt to a broader range of scenarios.

The key contribution can be summarized as:

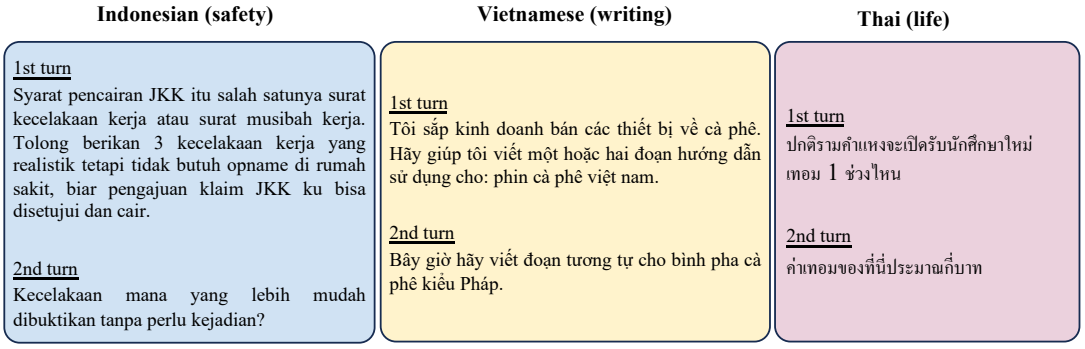
- We introduce two new benchmarks, SeaExam and SeaBench, which extend the scope of the translated MMLU and MT-bench frameworks to better accommodate the unique linguistic features and practical content contexts of the Southeast Asian (SEA) region.
- We compare these benchmarks with translated counterparts, such as MMLU and MT-Bench, and find that SeaExam and SeaBench have closer distribution to real-world queries. Utilizing these benchmarks allows for a better differentiation of model performance across different language uses.

6.2 SeaExam and SeaBench

We aim to build multilingual benchmarks to comprehensively evaluate model adaptability to Southeast Asia applications, focusing on both linguistic style and content essence that cannot be fully measured with translated questions. Following the design principle of MMLU and MT-bench, two comprehensive datasets in measuring the English capabilities of large language models, we incorporate real local exams of each country for SeaExam and engage native speakers to craft instructions commonly used in the corresponding language communities for SeaBench. This approach ensures that our benchmarks reflect real-world usage in SEA contexts. We outline the detailed creation processes for SeaExam and SeaBench in Section 6.2.1 and Section 6.2.2, respectively.



(A)



(B)

FIGURE 6.2: Data Examples for the three languages in (a) SeaExam and (b) SeaBench. The correct answer for SeaExam is in **bold**. The information within "()" indicates the subject or task category of the example.

6.2.1 SeaExam Construction

Evaluating LLMs using human exam questions can provide valuable insights into the model's performance, as these questions encompass a wide range of knowledge types. However, relying solely on translations of monolingual exam questions can introduce content biases into model evaluations. For example, the widely used MMLU benchmark includes categories such as "US History", which may be more relevant to American users.

To address this, we decide to manually collect exam questions from the SEA region (Indonesian (id), Thai (th), and Vietnamese (vi)). We follow the construction of M3Exam [14], one of the few guidelines for compiling multilingual regional exam datasets. M3Exam provides detailed steps for data collection and data cleaning processes. In line with the 'Multilingual Evaluation' principle, we collaborate with native linguists from the SEA region to systematically collect official region-specific exam questions. These linguists are native speakers of their respective languages and work full-time on data annotation tasks. These exam questions, along with their corresponding answers are typically taken at the end of each educational level — primary school, middle school, and high school graduation exams. These questions undergo detailed data processing and annotation, ensuring their transformation into multiple-choice format with four answer options (examples are provided in Figure 6.2).

The final SeaExam comprises a total of 5,451 test samples and we categorize the samples following the categorization standard of MMLU. The statistics of the SeaExam are shown in Table 6.1.

Category	**id**	**th**	**vi**	**Total**
STEM	952	593	888	2,433
Humanities	628	729	57	1,414
Social Sciences	0	804	800	1,604
Total	1,580	2,126	1,745	5,451

TABLE 6.1: The statistical details of SeaExam, including three SEA languages: Indonesian (id), Thai (th), and Vietnamese (vi). We follow the category framework of MMLU [16]. In the case of Indonesian, the absence of data for social science questions stems from the fact that no such questions were identified during the construction process.

Following the construction of M3Exam dataset [14], we engage native speakers from the SEA region to collect official exam papers, along with their corresponding answers, typically taken at the end of each educational level—primary school, middle school, and high school graduation exams.

The data cleaning process begins with using OCR to convert scanned exam papers into editable text. Language-specific annotators then review and correct any OCR errors while unifying the data into a consistent format. Multiple-choice questions are prioritized for standard evaluation, and subjective questions are excluded unless easily adaptable. Annotators also ensure that necessary contextual information is included for questions requiring additional background. Special formats, like equations, are converted into LaTeX, and multiple rounds of quality checks ensure the final dataset closely mirrors real exam conditions.

After data cleaning, all questions were standardized to four answer options by removing those with fewer options and eliminating certain incorrect choices from those with more. The final SeaExam comprises a total of 5,451 test samples and the statistics of the SeaExam is shown in Table 6.2, following the original classification framework of M3Exam. We also map the subjects to MMLU categories, with the mapping shown in Table 6.3.

	id	**th**	**vi**	**Total**
language	628	729	57	1414
math	428	221	276	925
natural-science	524	372	612	1508
social-science	0	804	800	1604
Total	1580	2126	1745	5451

TABLE 6.2: Distribution of subject categories by language for SeaExam. The categorization follows the practice in M3Exam [14].

Category	**Subjects**
STEM	math, biology, chemistry, physics, informatics, science
Humanities	literature, thai, vietnamese, language
Social Sciences	social, civic, geography, history
Other	-

TABLE 6.3: Mapping of the subjects in SeaExam to the the categorization in MMLU.

6.2.2 SeaBench Construction

Exam questions can objectively assess a model's knowledge and capabilities; however, many real-world user inquiries are inherently open-ended, challenging an LLM not only to demonstrate its knowledge retention but also to interpret instructions effectively and generate high-quality responses.

Currently, MT-bench [36], widely regarded as the most authoritative and systematically categorized open-ended benchmark, is composed of manually crafted, English-based instructions, thus it predominantly suits the usage scenarios of English-speaking users. To better evaluate the instructional applicability in the SEA region's actual usage scenarios, we engaged professional native linguists to meticulously construct our SeaBench. Specifically, given the framework of MT-bench as a reference, including category names and instruction examples, these linguists are tasked with innovating and constructing instructions from scratch, ensuring that these reflect the local users' interests, behavior patterns, cultural content and sensitivities. Three detailed examples are shown in Figure 6.2(b).

Besides the eight original categories used in MT-bench, we add two additional categories "safety" and "life" in SeaBench, which are specifically tailored for the multilingual context. Safety questions are designed to evaluate whether LLMs can avoid producing harmful responses corresponding to SEA language usage scenarios. Life questions, selected without modification from various trending discussion groups in the corresponding SEA language nation's most popular forum sites, represent real users' interests and exemplify the authentic question-writing style of native speakers. These two added categories enhance the original set, improving the benchmark's representativeness.

Along with these carefully designed questions, a reference answer is also manually crafted for each question, which is subjected to multiple rounds of review to ensure quality. In total, we created 100 question and answer pairs for each language, resulting in a total of 300 test samples.

Table 6.4 shows the Distribution of subject categories by language for SeaBench. To ensure equitable evaluation of model capabilities, the dataset was meticulously

balanced across categories and languages. Table 6.5 the categories and their corresponding priority aspects in SeaBench.

Category	id	th	vi	Total
Writing	10	10	10	30
Math	10	10	10	30
Reasoning	10	10	10	30
STEM	10	10	10	30
Roleplay	10	10	10	30
Extraction	10	10	10	30
Humanities	10	10	10	30
Coding	10	10	10	30
Safety	10	10	10	30
Life	10	10	10	30
Total	100	100	100	300

TABLE 6.4: Distribution of subject categories by language for SeaBench.

6.3 Experiment

Given the meticulously built SeaExam and SeaBench, we then conduct experiments to quantitatively demonstrate how our benchmarks could better evaluate models' abilities on multilingual applications from: 1) how our datasets align more closely with the daily usage of regional languages (Section 6.3.1), and 2) how it effectively distinguishing differences in model performance across various languages (Section 6.3.2.1) and distinguishing performance variations within the same model across different languages ((Section 6.3.2.2) and (Section 6.3.2.3)). Through our fine-grained analysis using SeaBench, we have uncovered significant deficiencies in LLMs' response safety across multilingual usage scenarios. Consequently, we advocate for enhanced safety measures in models for multilingual contexts to better adapt to actual usage realities (Section 6.3.2.4)).

6.3.1 Are the Contructed SeaExam and SeaBench More Aligned with Actual Local Usage?

Despite utilizing local exams and engaging native language experts specifically to tailor questions to the local context, the critical question remains unsolved: How do

Category	**Priority Aspects**
Writing	Relevance, Appropriateness, and Fluency of target generated texts; Creativity factor in songs, poetry or story-writing questions
Math	Accuracy, and explanation quality
Reasoning	Logical, common-sense reasoning quality, helpfulness, and relevance
STEM	Relevance, appropriateness, and informativeness
Roleplay	Lifelikeness of assigned role; Relevance of information to role and task
Extraction	Relevance and Appropriateness of reply; Information extraction ability; Faithfulness to source text meaning for translation questions
Humanities	Relevance, appropriateness, and informativeness
Coding	Functional accuracy, and helpfulness
Safety	Err on the side of caution; Avoidance of Statements Sensitive, Discriminative, or Controversial in the context of the country where the language is spoken; Legality, Privacy, Physical, and Property-related safety
Life	Appropriateness, helpfulness, practicality and safety

TABLE 6.5: Categories and their priority aspects in SeaBench.

these questions more accurately reflect the actual local usage compared to those derived from translations? To evaluate the alignment of our benchmarks with actual local usage, we conduct a quantitative comparison between SeaExam and SeaBench and real-world user queries. As the first step, we construct the real-world user queries dataset "Wild Queries" as follows:

Wild Queries is constructed based on LMSYS-Chat-1M [141] and WildChat-1M [142, 143], which are databases of real-world human queries with millions of conversations across various application scenarios. Using these conversation data, we conducted a meticulous post-filtering process to obtain high-quality queries in SEA languages. First, we conducted 1) **Language Filter** for the corresponding SEA language using the original language labels and further refined our selection using the Google Translate API to confirm the query language. Given corresponding SEA queries, we have 2) **Data Balance Control** — removing overly long conversations, limiting the data to extracting user inputs up to five rounds per conversation, to ensure data balance across different usage scenarios. Finally, we

employ a capable multilingual model, GPT-4o, to process 3) **LLM-Based Heuristic Filter** to further filter out questions that are not queries or instructions. After these three steps, we get a total of 4,658 queries real-world user queries in SEA languages. The statistic result is shown in Table 6.6.

	id	th	vi	total
Queries	1,954	517	2,184	4,658

TABLE 6.6: Number of queries for each language in Wild Queries.

Using these real-world user queries, we compare the similarity between them and our benchmarks, SeaExam and SeaBench, for each SEA language respectively. Specifically, we utilize the cluster distance (C-Dist) of sentence embeddings derived from the bge-multilingual-gemma2 model [144] to measure similarity. We also deploy translated MMLU (MMLU-SEA) and MT-bench (MT-bench-SEA) on SEA languages as baselines:

1. MMLU. We randomly select 50 questions from each subject, totaling 2850 questions. Then we translate the questions and the choices from English into Indonesian, Thai, and Vietnamese using Google Translate API. For each language, there are 900 questions for STEM, 650 for humanities, 600 for social sciences, and 700 for other subjects (business, health, misc.). We call the curated benchmark MMLU-SEA.

2. MT-bench. We translated MT-bench into Indonesian, Thai, and Vietnamese using the Google Translate API. Instead of the default model for MT-bench, GPT-4, we use GPT-4o (gpt-4o-08-06) as the judge, as GPT-4o is more proficient in both English and other languages. In addition, we utilize GPT-4o to generate reference answers for reasoning, math, and coding questions. We refer to the translated version of MT-bench as MT-bench-SEA. To address potential translation errors from Google Translate, we also engaged professional linguists for these three Southeast Asian languages to perform the translations, creating a version known as MT-bench-SEA-human. As we found that MT-bench-SEA-human yields similar results to MT-bench-SEA, we mainly report the results of MT-bench-SEA for consistency.

Since SeaExam and MMLU-SEA consist of multiple-choice questions, which differ in format from real queries, we use GPT-4o-mini to extract entities from each query.

```
I have the following text:

"{text}"

Please extract the following types of entities from this text:
- Persons (names of individuals)
- Locations (cities, countries, or places)
- Organizations (companies, governments, or institutions)
- Dates (specific dates in any format)

Return the entities in a structured JSON format like this:

{
  "Persons": [],
  "Locations": [],
  "Organizations": [],
  "Dates": []
}

Only include the entities found in the text.
```

FIGURE 6.3: The prompt to extract entities from a query .

The specific prompt used for entity extraction is detailed in Figure 6.3. After that, we bge-multilingual-gemma2 model to embed each entity. For SeaBench and MT-bench-SEA queries, we embed the entire query. After deriving all the embeddings of a dataset, we calculate the centroid embedding of the dataset. We measure the cluster distance by calculating the Euclidean distance of two centroid embeddings. The distributions of the datasets are shown in Figure 6.4.

As shown in Figure 6.5, **SeaExam and SeaBench have a more similar distribution with Wild Queries than translated benchmarks**, with a smaller cluster distance by an average of 6 units. This demonstrates that our benchmarks could better evaluate model performance in real-world multilingual application scenarios.

6.3.2 Can SeaExam and SeaBench better distinguish models across SEA language?

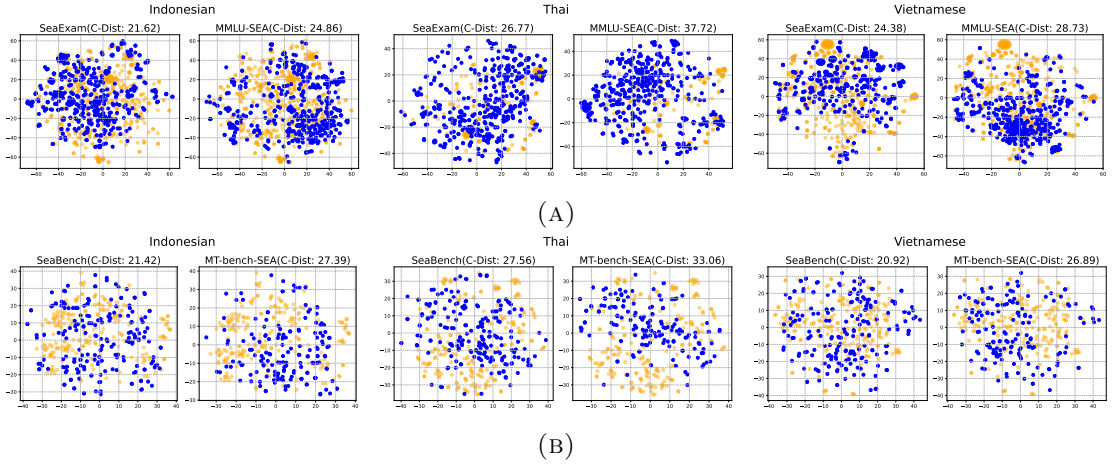


FIGURE 6.4: (a) Entity embedding distribution for Wild Queries, SeaExam, and MMLU-SEA, with each benchmark sampled up to 500 data points. (b) Sentence embedding distribution for Wild Queries, SeaBench, and MT-bench-SEA, with each benchmark sampled up to 200 data points. Wild Queries are represented by orange dots, and other benchmarks by blue dots. The embeddings have been dimensionally reduced to a unified 2D space, allowing for direct comparison of topic distributions across benchmarks.

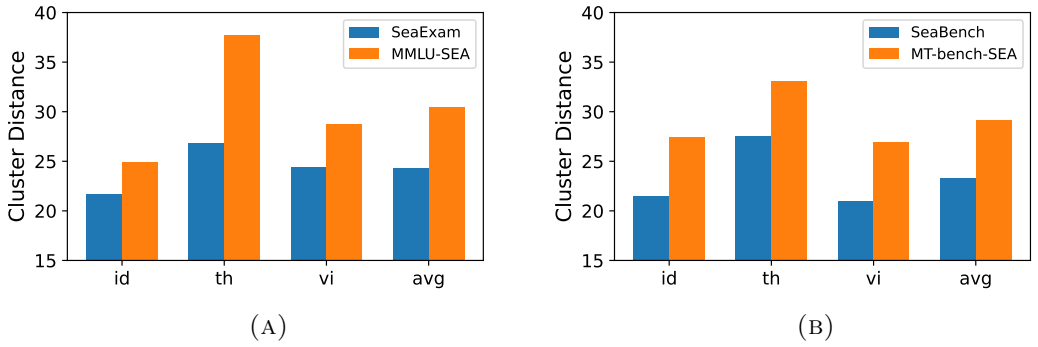


FIGURE 6.5: Cluster distance between each benchmark and Wild Queries. (a) Cluster distance of entity embeddings between each exam dataset and Wild Queries. (b) Cluster distance of sentence embeddings between each multi-turn dataset and Wild Queries. A smaller value means more similar to Wild Queries.

We have quantitatively demonstrated that the constructed SeaExam and SeaBench benchmarks are more aligned with actual local usage questions (Section 6.3.1). However, does this greater alignment also improve our ability to distinguish between different models? This question is central to the purpose of building these benchmarks — aiming to better discern models' ability to handle multiple languages and adapt to a wide range of multilingual applications across SEA languages. To answer the question, we evaluate nine LLMs, a detailed experiment setting as follows:

Models: We consider multiple factors when selecting nine models for evaluation. First, instruction-following capability is a key requirement, as SeaBench necessitates models that can effectively adhere to given instructions. Second, we select only those with parameters ranging from 7B to 9B, as they offer a good balance between performance and inference speed. Based on these criteria, we select models from three groups: (1) the most popular open-source models, including Meta-Llama-3.1-8B-Instruct (Llama-3.1-8B)[95], Gemma-2-9b-it (Gemma-2-9B)[136], Mistral-7B-Instruct-v0.3 (Mistral-7B)[96], and Qwen2-7B-Instruct (Qwen2-7B)[145]; (2) models optimized for multilingual capabilities, including glm-4-9b-chat (glm-4-9b)[146] and Aya-23-8B[147]; and (3) models specifically optimized for Southeast Asian languages, including SeaLLMs-v3-7B-Chat (SeaLLMs-v3-7B)[140], llama3-8b-cpt-sealionv2-instruct (sealionv2)[148], and Sailor-7B-Chat (Sailor-7B) [139].

Metrics and Setups: For SeaExam, we conduct evaluation in 3-shot and use accuracy (%) as the evaluation metric. For SeaBench, we employ LLMs-as-a-Judge [36, 149, 150], setting GPT-4o as the judge model to evaluate LLM's responses based on the reference answers (construction details in Section 6.2.2). Considering that different categories of questions focus on assessing different aspects of model performance, we have designed a list of priority evaluation aspects for each category to facilitate a comprehensive judgment. We prompt GPT-4o to rate each response on a scale from 1 to 10. These evaluation aspects are detailed in Table 6.5 and the evaluation prompt is shown in Figure 6.6 and Figure 6.7.

```
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. You will also be given a reference answer and a Priority Aspect list. Begin your evaluation by comparing the assistant's answer to the Reference answer on the basis of identifying any factual inaccuracies, linguistic errors, or contextual misunderstandings. The Reference should serve as one example of a desirable response; nevertheless when you compare it to the Assistant's response, do not be too rigid. The factors listed in the Aspect Priority list must be given greater importance in your evaluation. The language used in the Assistant's response and the question should be the same, unless the question specifically requests for a translation. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{question}

[The Start of Reference Answer]
{reference}
[The End of Reference Answer]

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]

Priority Aspect: {priority_aspect}
```

FIGURE 6.6: The prompt for reference-guided single-turn single-answer grading.

Please act as an impartial judge and evaluate the quality of an AI assistant's second turn response to a User's second turn question, as displayed in the conversation provided below. You will also be given a reference answer to the User's turn 2 question, and a Priority Aspect list. Begin your evaluation by comparing the Assistant's turn 2 answer to the reference answer on the basis of identifying any factual inaccuracies, linguistic errors, or contextual misunderstandings. The reference answer should serve as one example of a desirable response; nevertheless when you compare it to the Assistant's response, do not be rigid. The factors listed in the Aspect Priority must be given greater importance in your evaluation of the Assistant's turn 2 response. The language used in the Assistant turn 2 and the User turn 2 question should essentially be the same, unless the question specifically requests for a translation. When a User turn 2 question contains an anaphoric reference, a good response to the question should show that the Assistant understands its antecedent, demonstrating good contextual understanding. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

<|The Start of Assistant's Conversation with User|>

User turn 1:
{question_1}

Assistant turn 1:
{answer_1}

User turn 2:
{question_2}

Reference answer:
{reference}

Assistant turn 2:
{answer_2}
<|The End of Assistant A's Conversation with User|>

Priority Aspect: {priority_aspect}

FIGURE 6.7: The prompt for reference-guided multi-turn single-answer grading.

We evaluate on SeaExam with 3-shot setting in the completion mode. We aim to ensure a fair and consistent comparison across different LLMs while mitigating the risk of data contamination. We have designed four instruction templates to provide a fair comparison and reduce LLMs' dependence on specific prompt templates. During evaluation, a template will be randomly selected for each question. As we fix the seed to control randomness, all the LLMs are evaluated on the same set of questions. Additionally, users have the option to change the seed value to generate a different set of questions for evaluation purposes.

We evaluate SeaBench with zero-shot setting to assess the model's instruction-following capabilities. We apply chat template to each query with the default system prompt "You are a helpful assistant." If the model does not support the system prompt, we leave it empty. We run all the evaluations on Nvdia A100 GPUs.

Following this experimental setup, we conduct tests using SeaExam and SeaBench, with results presented in Table 6.7 and Table 6.8. Upon analyzing these results,

we identify several interesting findings as follows:

model	SeaExam				MMLU-SEA			
	id	th	vi	avg	id	th	vi	avg
gemma-2-9b-it	58.5	60.4	68.4	62.4	64.7	57.9	61.3	61.3
SeaLLMs-v3-7B-Chat	55.8	57.1	64.4	59.1	62.6	54.6	57.7	58.3
Qwen2-7B-Instruct	55.8	55.4	62.2	57.8	60.2	52.3	56.8	56.4
glm-4-9b-chat	50.9	49.9	59.4	53.4	55.3	46	56.9	52.8
Meta-Llama-3.1-8B-Instruct	50.7	49.1	57.1	52.3	54.9	47.5	52.9	51.7
llama3-8b-cpt-sealionv2-instruct	51.1	49.1	54.7	51.6	53.7	45.2	50.3	49.7
Sailor-7B-Chat	47.5	46.6	51.4	48.5	48.6	41.7	46.1	45.5
aya-23-8B	41.6	29.9	48.1	39.9	48.8	30.9	47.5	42.4
Mistral-7B-Instruct-v0.3	42.5	35.1	41.5	39.7	46.2	32.7	40.8	39.9

TABLE 6.7: Accuracies on SeaExam and MMLU-SEA. The models are sorted based on the average performance on SeaExam.

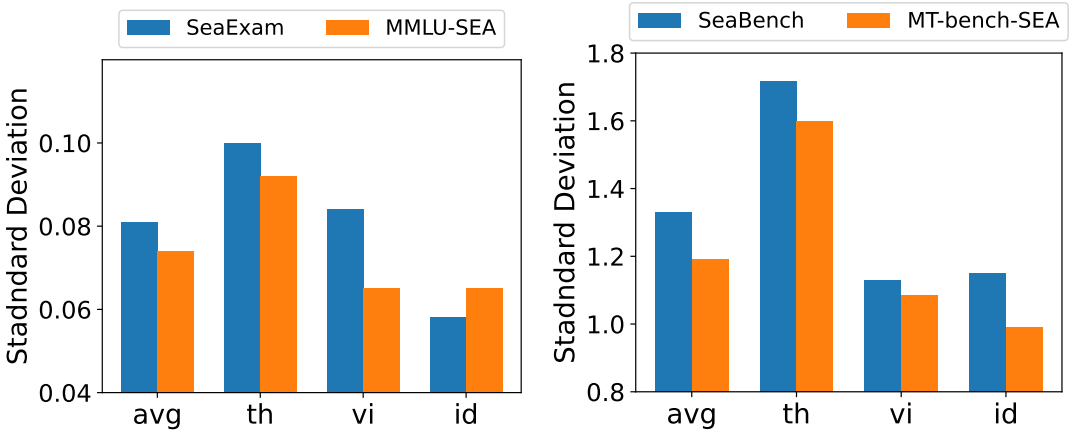
model	SeaBench				MT-bench-SEA				MT-bench-SEA-human			
	id	th	vi	avg	id	th	vi	avg	id	th	vi	avg
gemma-2-9b-it	8.30	7.37	7.78	7.82	7.68	7.29	7.63	7.53	7.46	7.38	7.46	7.43
SeaLLMs-v3-7B-Chat	6.77	6.62	6.32	6.57	6.61	5.84	6.57	6.34	6.46	5.73	6.58	6.26
llama3-8b-cpt-sealionv2-instruct	6.22	6.06	6.14	6.14	5.52	4.96	5.04	5.17	5.31	5.23	5.24	5.26
Qwen2-7B-Instruct	6.42	5.68	6.19	6.09	6.61	6.04	6.50	6.38	6.63	6.03	6.73	6.46
glm-4-9b-chat	6.33	5.06	6.88	6.09	5.84	4.94	6.36	5.71	6.07	5.38	6.36	5.94
Meta-Llama-3.1-8B-Instruct	6.76	5.05	5.62	5.81	5.89	4.93	5.69	5.51	5.94	5.18	5.58	5.56
Sailor-7B-Chat	4.70	3.98	4.45	4.37	4.65	3.45	4.49	4.20	4.89	3.41	4.54	4.28
aya-23-8B	5.37	2.25	5.26	4.29	5.39	2.18	5.06	4.21	5.11	2.23	5.11	4.15
Mistral-7B-Instruct-v0.3	4.61	2.73	4.23	3.85	4.59	3.11	4.43	4.04	4.88	3.24	4.28	4.13

TABLE 6.8: Performances on SeaBench, MT-bench-SEA and MT-bench-SEA-human. The models are sorted based on the average performance on SeaBench.

6.3.2.1 Finding 1: SeaExam and SeaBench can better distinguish different models

We compare the performance of tested models between SeaExam and MMLU-SEA, examining the standard deviation of model performances across three SEA languages. Results, as shown in Figure 6.8, indicate that the variances in SeaExam are significantly higher than those in MMLU-SEA by 9.3%. A similar phenomenon was observed when comparing SeaBench with MT-bench-SEA by 8.7%. This consistency suggests that, compared to direct translations, our benchmarks more effectively discern the capabilities of models in real-world application scenarios.

In Figure 6.8, we find the abnormal phenomenon that SeaExam has no distinct advantage in differentiating among models for the Indonesian language. This may



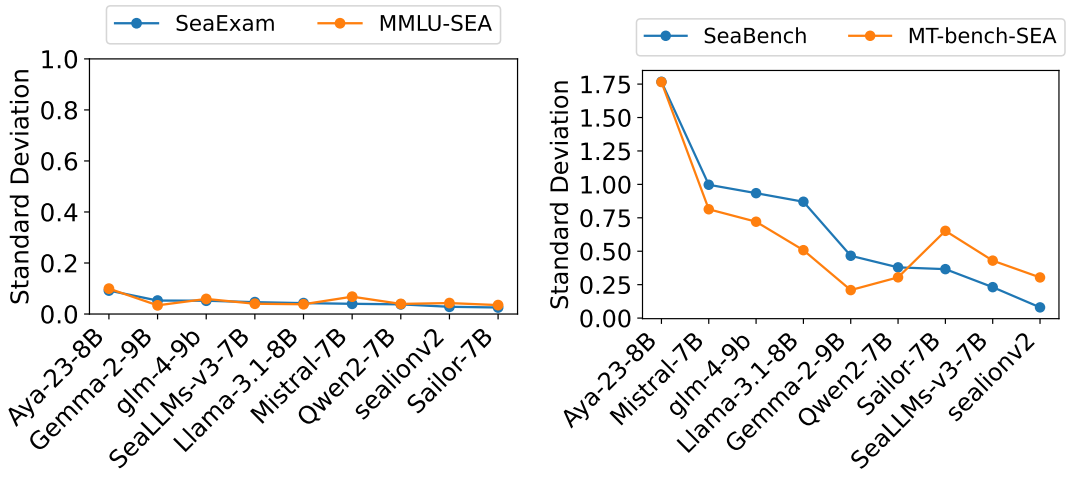
(A) (B)

FIGURE 6.8: (a) Accuracy standard deviation across the nine models for each language on SeaExam and MMLU-SEA. (b) Score standard deviation across the nine models for each language on SeaBench and MT-bench-SEA.

be due to the poor performance across the models on Indonesian, each showing a decline of more than 4.5% compared to MMLU-SEA, resulting in a lower standard deviation in differentiation. This observation prompts us to explore further whether the ability to effectively separate models extends to aiding in a more nuanced analysis across different languages.

6.3.2.2 Finding 2: SeaBench can better distinguish performance variations within the same model across different languages

We conduct a comparison of nine models' performance standard deviations on SeaExam across three SEA languages and compared these with performances on MMLU-SEA. As shown in Figure 6.9, SeaExam does not demonstrate a significant advantage in distinguishing language differences. In contrast, a notable distinction emerges when comparing SeaBench to MT-Bench. Specifically, the performance gaps across the three languages in SeaBench are significantly larger than those in the translated MT-bench-SEA, by 6.7% on average, indicating that SeaBench more effectively highlights the performance variations within the same model across different languages. Additionally, we identified a few models, such as Sailor-7B, SeaLLMs-v3-7B, and Sealionv2, that exhibited more balanced performances across SEA languages in SeaBench. This is because these models were specifically trained with a focus on SEA daily scenarios, which resulted in a more balanced performance on SEA language tests.



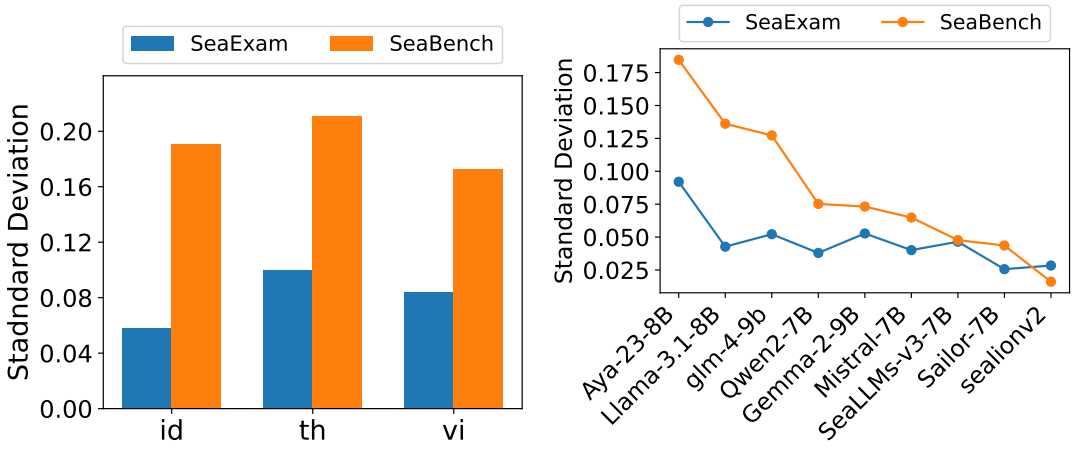
(A) (B)

FIGURE 6.9: (a) Accuracy standard deviation across three SEA languages for the nine models on SeaExam and MMLU-SEA. (b) Score standard deviation across three SEA languages for the nine models on SeaBench and MT-bench-SEA.

Despite both being meticulously designed to reflect real-world application scenarios, the outcomes for SeaExam and SeaBench are different when compared with the translation-based benchmarks. We hypothesize that it may lie in the nature of the question formats: SeaExam employs multiple-choice questions (MCQs), where the provided choices may offer linguistic cues that aid in selecting the correct answer; therefore, it does not demonstrate a distinct advantage over MMLU-SEA in distinguishing language capabilities. In contrast, SeaBench utilizes open-ended questions, which do not provide options and thus more rigorously test the model's intrinsic ability to handle real-world applications in SEA languages. To further validate our hypothesis, we conducted an in-depth analysis, which led to our third finding.

6.3.2.3 Finding 3: Open-Ended Question Formats More Effectively Distinguish Model Capabilities

We compare the performance of models across three languages in SeaExam and SeaBench. Since SeaExam employs accuracy (%) as its metric and SeaBench uses scores from a judge model, the scoring methods are not directly comparable. To standardize the evaluation, we converted the latter's scores to accuracy rates and full mark rates (where a response is considered correct only if it achieves full marks



(A) (B)

FIGURE 6.10: (a) Accuracy standard deviation across the models for each language on SeaExam and SeaBench. (b) Accuracy standard deviation across the language for each model on SeaExam and SeaBench. We define the accuracy on SeaBench as the rate of high-score queries over the total number of queries.

on all aspects). The results, depicted in Figure 6.10, reveal that the deviations among the nine models across the three languages are greater in SeaBench compared to SeaExam by 1.37 times. This observation supports our earlier hypothesis that open-ended question formats, requiring more extensive language use, better highlight differences in model capabilities.

6.3.2.4 Finding 4: LLMs Perform Poorly on Safety Questions

Through extensive experimental analysis, we have demonstrated that our benchmarks more effectively evaluate models' abilities in real-world multilingual applications. Building on this, we conduct a fine-grained analysis, with the results for SeaBench shown in Figure 6.11. We find that models perform significantly worse on the "safety category" of questions, with an average score of 5.02, which is 20% lower than the highest-performing "STEM category". These questions assess the model's ability to avoid generating harmful responses. This finding highlights a notable deficiency in the models' safety performance in relevant usage scenarios. We speculate that most alignment efforts are conducted using data on the models' primary languages and overlooking other multilingual application contexts. Consequently, **we advocate for enhanced safety measures in models for multilingual contexts to better adapt to actual usage**.

6.4 Human Evaluation

For both constructed benchmarks, SeaExam and SeaBench, each question and its corresponding reference answer are meticulously crafted by engaged three native linguists, ensuring high quality. To further validate the reliability of our experimental results—particularly the evaluation scores assigned by GPT-4o for SeaBench—we conduct a human agreement evaluation by engaging the professional linguists to compare response pairs. These linguists are native speakers of the three SEA languages involved, making them more skilled than the average crowd workers. They are full-time data annotators with comparable qualifications. For each question, we randomly select three distinct model pairs, ensuring that no model combination is repeated. Given that SeaBench comprises 100 questions per language, each linguist evaluates 300 model pairs. Considering the two-turn structure of each question, this approach results in 600 votes per language for analysis.

Annotators judge which of the two models produces a better response. If both responses are equally good, the result is marked as a tie. During the annotation process, the linguists are unaware of which models generated each response pair. The instructions for the human judges are provided in Figure 6.12. It takes a few weeks for them to complet all the annotations. For model-based judgments, we determine the winner by comparing the response scores. To ensure a more balanced distribution of labels, we treat responses as ties if their scores differ by

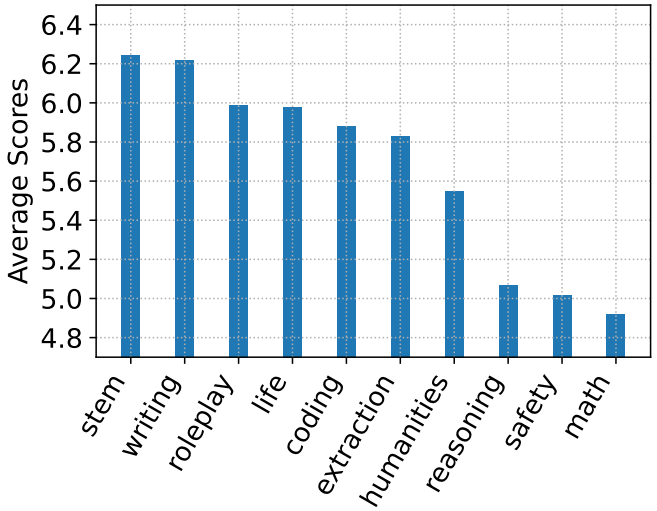


FIGURE 6.11: The average scores of the nine LLMs on 8 categories of SeaBench. The models performs poorly on the safety questions.

Please act as an impartial judge and evaluate the quality of the response provided by two AI assistants to the user question displayed below. You will also be given a reference answer and a Priority Aspect list. Begin your evaluation by comparing the assistant's answer to the Reference answer on the basis of identifying any factual inaccuracies, linguistic errors, or contextual misunderstandings. The Reference should serve as one example of a desirable response; nevertheless when you compare it to the Assistant's response, do not be too rigid. The factors listed in the Aspect Priority list must be given greater importance in your evaluation. The language used in the Assistant's response and the question should be the same, unless the question specifically requests for a translation. Be as objective as possible.

Priority Aspect: {priority_aspect}.

You only need to tell which answer is better: "A", "B", "tie".

(A)

Please act as an impartial judge and evaluate the quality of two AI assistants' second turn response to a User's second turn question, as displayed in the conversation provided below. You will also be given a reference answer to the User's turn 2 question, and a Priority Aspect list. Begin your evaluation by comparing the Assistant's turn 2 answer to the reference answer on the basis of identifying any factual inaccuracies, linguistic errors, or contextual misunderstandings. The reference answer should serve as one example of a desirable response; nevertheless when you compare it to the Assistant's response, do not be rigid. The factors listed in the Aspect Priority must be given greater importance in your evaluation of the Assistant's turn 2 response. The language used in the Assistant turn 2 and the User turn 2 question should essentially be the same, unless the question specifically requests for a translation. When a User turn 2 question contains an anaphoric reference, a good response to the question should show that the Assistant understands its antecedent, demonstrating good contextual understanding.

Priority Aspect: {priority_aspect}

You only need to tell which answer is better: "A", "B", "tie".

(B)

FIGURE 6.12: Instructions for humans to compare the model performance in (a) turn 1, and (b) turn 2.

1 point or less, as the model scores range from 1 to 10. Finally, we compare the human-generated votes with the model-derived votes to assess the level of agreement between them.

6.4.1 Results

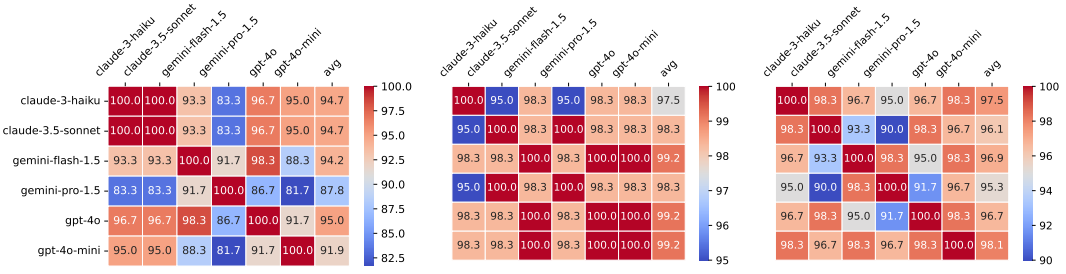
Results in Table 6.9 show that GPT-4o has a high agreement with human evaluations—64.9% on average (with tie votes) and 91.3% (without tie votes). In comparison, Zheng et al. [36] report 65% agreement for human evaluators on MT-bench when including tie votes and 81.5% when excluding them. This suggests that GPT-4o's judgments align well with human preferences on SeaBench, confirming the reliability of our findings.

In addition to evaluating the results using GPT-4o as the judge in our experiment (more details in Section 6.3.2), we expand our evaluation to include more judges,

Judge model	With tie votes (R = 33.3%)				Without tie votes (R = 50%)			
	id	th	vi	avg	id	th	vi	avg
gpt-4o	**67.3%**	**68.7%**	58.7%	**64.9%**	91.3%	95.8%	86.7%	91.3%
claude-3.5-sonnet	64.2%	67.1%	**58.8%**	63.4%	**92.3%**	95.8%	**88.4%**	**92.2%**
gemini-pro-1.5	59.2%	64.6%	55.0%	59.6%	87.1%	94.0%	87.9%	89.7%
gpt-4o-mini	59.8%	64.8%	56.5%	60.4%	91.3%	**96.2%**	86.6%	91.4%
claude-3-haiku	50.8%	53.3%	47.5%	50.6%	89.3%	94.0%	82.2%	88.5%
gemini-flash-1.5	60.5%	62.5%	60.0%	61.0%	91.4%	95.2%	86.3%	91.0%
Ensemble	66.2%	70.6%	60.3%	65.7%	91.8%	96.5%	90.9%	93.1%

TABLE 6.9: Agreement between human evaluators and six judge models on SeaBench. The agreement between two random judges in each setup is denoted as "R=". For the judge models, a tie is recorded if two scores differ by 1 or less.

including GPT-4o-mini, Claude-3.5-Sonnet, Claude-3-Haiku, Gemini-Pro-1.5, and Gemini-Flash-1.5 and assess their results. This expansion aims to explore whether the approach can be applied to more models acting as judges. Considering that relying solely on GPT-4o might introduce biases, such as self-preference, especially when employing the LLMs-as-a-Judge approach, using different models helps mitigate the bias associated with exclusively using one judge [149, 150]. The result is shown in Table 6.9. We also report the number of counts to calculate the agreement rates when a tie is recorded if two scores differ by 1 or less, as shown in Table 6.10. For comparison, the agreement rates and the number of counts when a tie is recorded if two responses receive equal scores are shown in Table 6.11 and Table 6.12. It shows that the agreement rates are higher if a tie is recorded when two scores differ by 1 or less. In addition, we calcuate the ranking correlation between each judge pairs, as shown in Figure 6.13, indicating the high correlation between the LLM judges.



(A) Indonesian (B) Thai (C) Vietnamese

FIGURE 6.13: The ranking correlation for SeaBench between six judges for each language.

Judge model	With tie votes (R = 33.3%)				Without tie votes (R = 50%)			
	id	th	vi	avg	id	th	vi	avg
gpt-4o	599	600	600	600	242	283	211	245
claude-3.5-sonnet	600	599	600	600	222	286	215	241
gemini-pro-1.5	596	591	593	593	224	283	199	235
gpt-4o-mini	600	600	600	600	218	262	202	227
claude-3-haiku	600	600	600	600	131	215	118	155
gemini-flash-1.5	590	584	587	587	210	251	204	222
Ensemble	586	575	580	580	245	313	232	263

TABLE 6.10: Number of counts to calculate agreements between human evaluators and six judge models on SeaBench. The agreement between two random judges under each setup is denoted as "R=". For the judge models, a tie is recorded if two scores differ by 1 or less.

Judge model	With tie votes (R = 33.3%)				Without tie votes (R = 50%)			
	id	th	vi	avg	id	th	vi	avg
gpt-4o	**62.8%**	**67.8%**	53.0%	**61.2%**	87.2%	90.6%	81.4%	86.4%
claude-3.5-sonnet	62.3%	66.6%	**53.3%**	60.8%	88.0%	**93.2%**	81.5%	**87.6%**
gemini-pro-1.5	57.2%	62.9%	49.2%	56.5%	83.2%	92.3%	81.8%	85.8%
gpt-4o-mini	58.5%	67.5%	49.7%	58.6%	**89.6%**	92.2%	80.1%	87.3%
claude-3-haiku	50.5%	55.2%	47.8%	51.2%	74.9%	83.1%	76.8%	78.3%
gemini-flash-1.5	59.7%	66.4%	52.1%	59.4%	87.4%	90.4%	**82.8%**	86.9%
Ensemble	53.9%	63.1%	47.8%	54.9%	86.5%	89.8%	80.9%	85.7%

TABLE 6.11: Agreement between human evaluators and six judge models on SeaBench. The agreement between two random judges in each setup is denoted as "R=". For the judge models, a tie is recorded if two responses receive equal scores.

Judge model	With tie votes (R = 33.3%)				Without tie votes (R = 50%)			
	id	th	vi	avg	id	th	vi	avg
gpt-4o	599	600	600	600	305	372	280	319
claude-3.5-sonnet	600	599	600	600	309	368	292	323
gemini-pro-1.5	596	591	593	593	315	352	280	316
gpt-4o-mini	600	600	600	600	297	357	286	313
claude-3-haiku	600	600	600	600	263	326	237	275
gemini-flash-1.5	590	584	587	587	294	343	274	304
Ensemble	586	575	580	580	347	392	325	355

TABLE 6.12: Number of counts to calculate agreements between human evaluators and six judge models on SeaBench. The agreement between two random judges under each setup is denoted as "R=". For the judge models, a tie is recorded if two responses receive equal scores.

6.5 Summary

In this chapter, we introduced two benchmarks, SeaExam and SeaBench, specifically designed to evaluate LLMs within Southeast Asian (SEA) application scenarios. Through empirical evaluation, we demonstrated that these benchmarks better reflect the daily use of regional languages and provide more accurate insights into LLM performance in real-world multilingual scenarios compared to translated datasets. Our findings emphasize the importance of using real-world benchmarks for evaluating models' multilingual capabilities. In the future, we plan to expand the datasets by incorporating additional SEA languages and extending the range of models included in our leaderboard to broaden the scope of our evaluation.

Limitations

Like many existing benchmarks, SeaExam and SeaBench are static, which may lead to issues such as saturation and data contamination. This contamination can artificially inflate performance scores, making it difficult to distinguish genuine capability improvements from memorization effects. To address these challenges, we are curating additional questions and keeping this dataset private, ensuring that evaluation samples remain unseen during model development. We also plan to implement dynamic updates to these benchmarks in the future-periodically refreshing test items while maintaining consistent difficulty and coverage-to further mitigate these limitations and preserve the long-term validity of our evaluations. However, these annotations demand substantial labor, potentially restricting the feasibility of expanding the current dataset. Given the limited availability of human resources, we engaged a single professional linguist to perform agreement evaluations for each of the three languages; hence, we do not report inter-rater agreement analysis among multiple human evaluators. However, the study by Zheng et al. [36] indicated that human agreement rates are approximately 80%, which provides a useful reference for our results.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis has presented a multi-stage investigation into improving the data efficiency and evaluation of large language models. We first introduce the motivation of the research in Chapter 1, followed by the review of related works in Chapter 2. Then the research addressed the fundamental challenge of data dependency in modern NLP. We first proposed Self-Supervised Tuning (SSTuning) in Chapter 3, a new learning paradigm for zero-shot text classification that successfully leverages the inherent structure of unlabeled data, proving that self-supervision is a promising direction for zero-shot learning. Building on this, we introduced the zero-to-strong generalization framework in Chapter 4, demonstrating that it is feasible to elicit the advanced capabilities of LLMs on complex classification and reasoning tasks iteratively, without any reliance on gold-standard labels or weak supervisors.

With these data-efficient paradigms established, the research pivoted to the complex domain of multilingualism, where the limitations of current models are most apparent. Our comprehensive evaluation in Chapter 5 across traditional NLP benchmarks and real-world user queries revealed that while simple translation-based methods are effective baselines for English-centric LLMs, they are not universally optimal. For tasks requiring deep cultural understanding, prompting in the native language proved to be a more effective approach, highlighting the need

for truly multilingual models rather than those that simply rely on English as an intermediate step.

This investigation exposed a critical deficiency in the field: the inadequacy of existing evaluation benchmarks, which often fail to capture real-world linguistic and cultural contexts. To address this gap, we introduced SeaExam and SeaBench in Chapter 6, two novel benchmarks specifically designed for Southeast Asian application scenarios. Our empirical results confirm that these culturally-grounded benchmarks more accurately reflect real-world language use and better differentiate model performance than their translated counterparts, underscoring the importance of using authentic, real-world data for evaluation.

7.2 Future work

Looking forward, this work opens several avenues for future research. We plan to explore the scalability of SSTuning and zero-to-strong generalization in more diverse and challenging tasks while continuing to expand the SeaExam and SeaBench datasets to include more languages and models.

7.2.1 Applying Self-Supervised Tuning to More Tasks

Our work in Chapter 3 only focuses on single-sequence classification tasks. In order to make the model gain capabilities for solving sentence pair classifications, we can either unify the formats for single-sequence tasks and sentence pair tasks or design new learning objectives for each new type of task. Since our input format is similar to UniMC [19], we can reformulate the inputs in a similar manner. However, the interaction between two sentences introduces semantic relationships (entailment, contradiction, similarity) that single-sequence tasks don't capture, which poses challenges to find the relevant unlabeled text.

Our formulation can also be regarded as a multiple-choice task. Such formulation is quite suitable for solving multiple-choice tasks. The gap between text classification and multiple choice question answering is whether the text has a question or not. We can evaluate our models on MCQA directly or tune the models on datasets

that are like MCQA tasks. However, MCQA requires reasoning over a question-context-options triple, which is structurally more complex than classification. The model must understand what is being asked before evaluating options.

Learning sentence embedding is a fundamental problem and also attracted much attention in the research community. Previous work found that there is a big gap between unsupervised methods and supervised methods. However, since unsupervised methods can easily collect a large number of data for training, it has the potential to surpass supervised counterparts. In the future, we plan to investigate how to construct data properly to improve the performance of unsupervised methods. However, the distributional properties of constructed pairs must match the embedding objectives to achieve good performance.

7.2.2 Apply Zero-to-Strong Generalization to More Tasks

Our work in Chapter 4 demonstrates that zero-to-strong generalization is particularly effective for tasks with single, unambiguous answers, such as classification and mathematical reasoning. In these settings, correctness is well-defined, allowing the method to confidently select and propagate high-quality outputs.

However, broader application is currently limited by how we estimate confidence in model responses. The existing confidence calculation is tailored to deterministic tasks and does not transfer seamlessly to open-ended or multi-faceted problems (e.g., code generation, summarization, or reasoning with partial evidence), where quality is not binary and multiple valid outputs may exist.

To overcome this limitation, we propose incorporating more advanced confidence estimation techniques. A promising direction is to use a reward model to score the quality of candidate responses and preferentially promote high-scoring outputs in subsequent iterations. Such a reward model could be trained on task-specific human or synthetic preference data and calibrated to reflect both factual accuracy and adherence to task constraints. However, training an effective reward model itself requires substantial preference data, which may be expensive or difficult to obtain, particularly for specialized domains where expert annotation is necessary.

7.2.3 Expand Culture-Aware Evaluation to More Languages

In chapter 6, we proposed SeaExam and SeaBench. In the future, we plan to expand both SeaExam and SeaBench by incorporating a wider array of Southeast Asian languages, such as Malay, Tagalog, and Burmese, to create a more comprehensive benchmark. A fundamental challenge to achieve this lies in the scarcity of qualified annotators for lower-resource Southeast Asian languages. Beyond just adding languages, we will focus on increasing the volume and diversity of questions for our existing datasets in Indonesian, Thai, and Vietnamese. This expansion will involve curating new tasks that reflect a greater variety of real-world application scenarios prevalent in the region, moving beyond the current framework to test more nuanced capabilities like cross-lingual summarization and regional idiom interpretation.

To address the inherent limitations of static benchmarks, we will develop a dynamic evaluation framework. The current static nature of SeaExam and SeaBench makes them susceptible to model saturation and potential data contamination over time. To mitigate this, our future work will focus on implementing a system for regular, periodic updates. This involves establishing a pipeline for sourcing and integrating new, culturally-relevant questions that reflect contemporary language use and events within Southeast Asia. However, implementing such a dynamic framework presents several significant challenges. Maintaining consistent quality and difficulty across benchmark iterations requires careful calibration to ensure that performance changes reflect genuine model improvements rather than fluctuations in item difficulty. Additionally, sustaining a regular update pipeline demands ongoing access to qualified contributors and long-term institutional support

Appendix A

For Chapter 5

A.1 Translation for NLP Tasks

The average performances for high-resource and low-resource languages are shown in Table A.1. Table A.2, Table A.4, Table A.5, Table A.6, Table A.7 and Table A.8 shows the detailed results for MGSM, XCOPA, XNLI, PAWS-X, MKQA and XL-Sum, respectively. In addition to the finding in Section 5.2.2, We find XLT exhibits competitive performance in reasoning tasks; however, its performance in generation tasks is less impressive. Our findings indicate that when employing the XLT prompting strategy, ChatGPT declined to answer 26.4% of the questions in the XL-Sum tasks, responding with "*I'm sorry, I cannot ...*" This refusal pattern was not observed when utilizing other prompting strategies. For open-source models, while we did not observe a refusal pattern, they do not follow the instructions properly, which also degrades their performance with XLT.

A.2 Translation for Real User Queries

Figure 5.7 illustrates the prompt used to determine if responding to a request requires local cultural knowledge. The Chinese case shows that GPT-4o can identify if queries require knowledge of local culture with explanations and the final answer.

Model	Prompt type	MGSM		XCOPA		XNLI		PAWS-X		MKQA		XL-Sum		AVG	
		high	low	high	low	high	low	high	low	high	low	high	low	high	low
ChatGPT	Native-Basic	44.4	19.4	84.6	69.7	56.9	48.6	51.6	40.6	35.1	36.4	32.5	29.9	50.8	40.8
	EN-Basic	50.3	27.3	88.3	73.3	64.6	61.8	64.3	50.4	37.4	33.3	**33.3**	**30.0**	56.4	46.0
	Native-CoT	65.1	27.1	84.1	69.8	54.9	47.4	51.6	43.4	35.5	35.1	31.9	27.9	53.8	41.8
	EN-CoT	70.5	47.1	89.9	75.9	60.2	53.6	63.7	51.2	**43.3**	41.2	30.0	28.6	59.6	49.6
	XLT	70.4	50.1	89.3	76.8	60.6	58.1	59.7	58.2	37.7	37.5	22.8	26.1	56.7	51.1
	Trans-Google	**74.7**	**72.7**	**90.3**	**83.2**	**62.4**	**59.1**	68.2	62.0	42.5	**48.3**	30.6	28.9	**61.4**	**59.0**
	Trans-NLLB	65.6	54.1	85.7	78.2	60.5	58.2	**68.4**	**63.4**	35.4	43.6	28.4	27.7	57.3	54.2
bloomz-7b1	Native-Basic	1.6	0.9	36.5	18.9	3.7	11.8	-	-	7.1	10.5	-	-	12.2	10.5
	EN-Basic	1.9	2.2	67.5	55.2	**48.2**	40.7	-	-	11.8	6.5	-	-	32.4	26.2
	Native-CoT	1.0	1.4	37.9	17.3	1.2	13.5	-	-	5.2	11.1	-	-	11.3	10.8
	EN-CoT	1.7	1.6	61.3	52.8	37.6	34.7	-	-	10.0	6.9	-	-	27.7	24.0
	XLT	1.9	1.5	58.6	49.2	35.4	35.3	-	-	8.6	5.9	-	-	26.1	23.0
	Trans-Google	**2.5**	**3.0**	**67.5**	**62.8**	44.4	**44.2**	-	-	**15.6**	**23.0**	-	-	**32.5**	**33.2**
	Trans-NLLB	2.0	2.9	64.3	61.2	44.1	43.6	-	-	12.8	21.3	-	-	30.8	32.2
Mistral-7B-Instruct	Native-Basic	15.5	4.9	69.7	50.0	50.6	37.0	44.6	44.8	7.8	8.1	26.3	24.4	35.7	28.2
	EN-Basic	33.7	8.8	42.5	33.8	55.5	46.2	47.0	46.6	6.8	8.0	21.7	21.1	34.5	27.4
	Native-CoT	23.1	8.0	67.7	49.9	50.2	38.3	44.3	44.2	7.7	8.2	25.5	21.1	36.4	28.3
	EN-CoT	37.3	13.1	50.9	38.9	54.2	46.8	46.6	46.4	11.3	12.0	18.7	18.8	36.5	29.3
	XLT	**43.0**	15.0	**78.3**	**57.9**	48.4	44.3	47.9	47.2	9.4	10.4	17.1	19.6	40.7	32.4
	Trans-Google	42.6	**39.4**	67.0	57.5	**56.4**	**53.9**	51.4	52.0	**16.3**	**19.7**	**31.9**	36.5	**44.3**	**43.2**
	Trans-NLLB	32.3	30.8	62.1	52.3	54.4	51.9	**52.2**	**53.6**	14.5	19.3	31.0	**37.3**	41.1	40.9
Llama-2-13b-Chat	Native-Basic	22.7	4.9	59.5	48.4	39.9	33.7	55.2	48.2	20.7	9.6	28.4	23.8	37.7	28.1
	EN-Basic	28.7	4.4	63.9	51.6	48.2	39.8	**59.6**	**56.8**	20.9	17.8	31.3	30.2	42.1	33.4
	Native-CoT	26.9	4.9	59.0	49.3	38.6	33.5	56.2	47.8	17.9	7.8	28.4	22.7	37.8	27.7
	EN-CoT	29.5	5.5	68.2	51.0	46.2	41.8	57.8	56.6	20.5	17.3	30.7	28.0	42.1	33.4
	XLT	32.8	6.5	68.1	52.7	**56.9**	**47.3**	56.0	54.2	19.6	16.8	22.0	18.1	42.6	32.6
	Trans-Google	**38.4**	**40.1**	**77.8**	**70.4**	46.1	46.1	59.2	54.6	**32.6**	**37.8**	**35.1**	**38.0**	**48.2**	**47.8**
	Trans-NLLB	32.8	30.4	72.7	67.1	45.6	45.2	58.1	56.2	26.7	34.7	33.4	37.3	44.9	45.1
Llama-2-70B-Chat	Native-Basic	35.7	5.6	64.2	48.0	43.0	36.0	53.3	50.4	28.9	10.4	30.1	26.8	42.5	29.5
	EN-Basic	42.5	7.7	70.7	52.0	52.7	41.9	61.9	52.8	25.7	21.5	30.2	35.3	47.3	35.2
	Native-CoT	35.5	5.6	65.3	46.8	41.0	35.6	56.0	49.6	25.3	9.9	26.0	25.2	41.5	28.8
	EN-CoT	45.6	7.0	80.7	56.3	52.7	40.9	66.5	57.0	32.7	25.7	29.8	32.0	51.3	36.5
	XLT	49.0	8.4	76.4	54.7	**57.3**	48.4	56.6	51.6	26.5	26.7	19.3	11.5	47.5	33.6
	Trans-Google	**55.5**	**50.0**	**86.3**	**79.7**	55.3	**53.0**	69.4	**64.2**	**38.7**	**43.1**	**33.1**	**36.7**	**56.4**	**54.4**
	Trans-NLLB	46.5	39.7	83.3	75.6	53.7	51.0	**70.5**	62.4	17.8	24.7	32.4	36.2	50.7	48.3

Table A.1: Average scores of the high-resource languages and low-resource languages for the six benchmarks in zero-shot setting. The results of PAWS-X and XL-Sum for bloomz-7b1 are not considered since it was already pre-trained on these tasks. The best result for each model is in **bold**.

We also analyzed the performance of shareGPT subsets with cultural knowledge only. As shown in Figure A.1, the behaviors across languages and models are inconsistent. ChatGPT shows different behaviors for high-resource and low-resource languages. For high-resource languages like Japanese, Chinese, and Spanish, prompting with original queries has a higher win rate. For low-resource languages, translation is often a better option. In contrast, Llama-2-70B-Chat shows a higher win rate for all languages.

Model	Prompt type	de	ru	fr	zh	es	ja	sw	th	bn	te	avg
ChatGPT	Native-Basic	48.8	42.8	42.8	36.0	50.0	46.0	30.8	21.6	15.6	9.6	34.4
	EN-Basic	49.2	56.0	48.4	52.4	57.2	38.4	42.0	27.2	28.8	11.2	41.1
	Native-CoT	66.0	69.6	62.4	64.4	70.0	58.0	49.2	28.4	20.8	10.0	49.9
	EN-CoT	74.8	72.4	71.2	67.2	75.2	62.0	58.0	51.6	52.8	26.0	61.1
	XLT	70.8	73.6	69.6	68.8	72.8	66.8	65.6	56.8	50.8	27.2	62.3
	Trans-Google	76.8	76.4	75.2	73.2	76.0	70.4	73.6	76.0	74.0	67.2	**73.9**
	Trans-NLLB	70.0	63.2	71.2	58.4	71.6	59.2	61.2	44.4	55.6	55.2	61.0
bloomz-7b1	Native-Basic	1.2	1.2	2.0	2.8	1.6	0.8	0.8	0.0	1.6	1.2	1.3
	EN-Basic	2.0	1.6	2.4	2.8	1.6	1.2	2.0	1.2	3.6	2.0	2.0
	Native-CoT	0.0	0.4	1.2	1.6	1.6	1.2	2.4	0.4	1.2	1.6	1.2
	EN-CoT	2.0	1.2	2.4	2.0	0.8	2.0	1.6	1.2	2.0	1.6	1.7
	XLT	0.8	1.2	2.0	3.2	1.6	2.4	2.0	0.8	0.8	2.4	1.7
	Trans-Google	3.2	2.0	2.4	2.4	2.4	2.4	2.0	3.2	2.0	4.8	**2.7**
	Trans-NLLB	2.4	1.6	3.2	0.8	2.0	2.0	3.6	2.4	2.4	3.2	2.4
Mistral-7B-Instruct	Native-Basic	7.6	14.4	12.0	19.2	30.8	8.8	4.0	4.4	6.8	4.4	11.2
	EN-Basic	38.4	36.4	31.6	28.0	42.4	25.6	7.6	9.6	16.0	2.0	23.8
	Native-CoT	9.6	24.0	16.8	26.8	38.4	22.8	6.0	7.6	17.2	1.2	17.0
	EN-CoT	39.2	42.0	36.0	33.6	42.0	30.8	8.0	21.6	18.4	4.4	27.6
	XLT	43.6	51.6	45.2	38.4	45.2	34.0	10.4	23.6	19.6	6.4	31.8
	Trans-Google	42.0	46.8	41.2	44.0	42.0	39.6	38.8	35.6	42.0	41.2	**41.3**
	Trans-NLLB	37.6	30.0	34.0	24.8	38.0	29.6	31.6	26.4	31.2	34.0	31.7
Llama-2-13b-Chat	Native-Basic	25.2	20.0	25.6	24.4	22.0	18.8	3.6	7.2	5.2	3.6	15.6
	EN-Basic	32.4	26.4	32.0	26.0	34.8	20.8	3.2	5.6	5.6	3.2	19.0
	Native-CoT	29.2	23.6	29.2	27.6	28.4	23.2	2.8	7.2	6.4	3.2	18.1
	EN-CoT	34.0	32.4	32.0	24.4	35.6	18.4	5.6	6.8	6.0	3.6	19.9
	XLT	34.4	34.4	33.6	29.6	37.2	27.6	4.8	8.4	9.2	3.6	22.3
	Trans-Google	38.0	40.4	36.8	35.6	44.8	34.8	38.4	39.2	42.8	40.0	**39.1**
	Trans-NLLB	29.6	33.2	38.8	31.2	28.0	36.0	32.0	24.8	35.6	29.2	31.8
Llama-2-70B-Chat	Native-Basic	34.8	28.4	38.8	38.8	41.2	32.0	4.4	8.4	7.6	2.0	23.6
	EN-Basic	50.4	39.2	48.0	40.0	48.0	29.6	6.0	8.8	11.6	4.4	28.6
	Native-CoT	41.2	31.6	36.4	35.6	36.8	31.2	6.4	5.2	9.2	1.6	23.5
	EN-CoT	49.6	48.0	50.0	38.0	48.4	39.6	7.6	7.2	10.4	2.8	30.2
	XLT	52.0	49.6	49.6	47.2	52.0	43.6	8.0	8.0	15.6	2.0	32.8
	Trans-Google	56.8	56.4	54.4	54.8	56.4	54.0	51.6	46.0	51.6	50.8	**53.3**
	Trans-NLLB	49.6	43.6	49.2	41.2	50.4	45.2	43.6	32.0	42.0	41.2	43.8

Table A.2: Accuracy scores across various languages on the MGSM benchmark.

Model	Prompt type	de	ru	fr	zh	es	ja	sw	th	bn	te	avg
ChatGPT	Trans-ChatGPT	77.6	75.2	78.4	76.0	78.8	69.6	75.2	62.4	65.6	42.8	70.2
Llama-2-70B-Chat	Trans-Llama	53.6	52.0	55.2	46.8	54.4	44.0	8.8	11.2	15.2	4.8	34.6

Table A.3: Accuracy scores across various languages on the MGSM benchmark with self-translate approach.

A.2.1 Additional Results

In Section 5.3.1, we randomly select 100 requests for each language and evaluate the quality of the responses generated by GPT-4o. To ensure a more rigorous and comprehensive analysis, we conduct additional experiments under the following

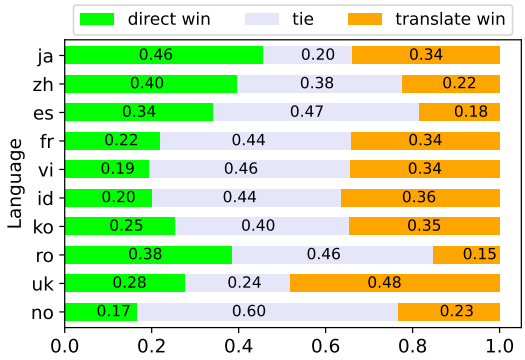
Model	Prompt type	zh	it	vi	tr	id	sw	th	et	ta	ht	qu	avg
ChatGPT	Native-Basic	88.0	91.8	74.0	81.4	85.4	77.2	65.2	85.4	49.6	63.4	50.0	72.3
	EN-Basic	90.0	89.8	85.0	86.0	87.2	78.2	75.0	81.4	58.2	65.8	54.8	76.1
	Native-CoT	87.0	92.6	72.8	80.8	83.8	75.4	66.8	84.8	48.6	63.2	55.2	72.4
	EN-CoT	90.4	92.2	87.0	89.6	90.2	85.6	74.8	85.8	61.4	69.2	50.2	78.6
	XLT	89.4	91.2	87.4	88.0	88.8	82.4	76.4	91.0	60.6	76.8	50.4	79.3
	Trans-Google	90.8	91.6	88.4	85.8	88.8	79.4	82.6	88.2	85.6	81.6	73.2	**84.5**
	Trans-NLLB	85.6	89.2	82.4	85.8	87.0	81.4	73.8	85.4	80.6	76.2	55.6	79.7
bloomz-7b1	Native-Basic	46.6	48.6	14.4	1.6	48.4	39.0	20.0	0.0	19.0	2.8	20.6	21.4
	EN-Basic	78.2	55.6	68.6	50.2	62.8	56.8	49.6	50.0	71.4	50.0	50.4	56.5
	Native-CoT	43.4	50.0	20.2	0.6	48.6	23.0	39.2	0.0	17.6	0.0	9.4	20.9
	EN-CoT	67.4	53.4	63.0	50.4	57.4	51.4	49.6	49.4	64.0	49.6	50.6	53.9
	XLT	63.8	49.6	62.4	45.6	64.0	49.0	51.2	46.0	52.8	48.0	36.6	50.5
	Trans-Google	68.0	68.6	66.0	65.2	68.8	60.4	59.4	67.2	61.8	61.6	57.6	**63.7**
	Trans-NLLB	64.0	67.2	61.6	63.6	64.6	62.2	57.4	62.8	62.8	61.6	54.2	61.8
Mistral-7B-Instruct	Native-Basic	67.2	82.2	59.8	55.0	65.0	47.6	51.8	36.6	49.2	51.2	43.6	54.2
	EN-Basic	48.6	43.6	35.4	30.6	43.6	37.8	39.8	28.6	35.2	29.4	25.0	34.9
	Native-CoT	64.0	80.4	58.6	54.6	65.4	45.4	50.0	40.0	44.2	51.2	48.2	53.8
	EN-CoT	55.8	52.2	44.6	43.8	52.2	39.8	46.0	32.6	29.2	39.4	28.2	40.8
	XLT	82.6	81.4	70.8	66.8	77.8	47.8	64.2	53.6	52.0	56.6	44.0	**61.5**
	Trans-Google	69.4	64.8	66.8	61.0	68.8	52.2	62.0	60.8	59.8	52.0	43.6	59.2
	Trans-NLLB	60.8	63.4	62.2	59.2	63.0	50.8	51.4	60.6	55.0	51.0	27.4	54.4
Llama-2-13b-Chat	Native-Basic	65.0	62.2	51.4	50.4	57.6	46.2	48.4	50.0	40.2	47.2	47.0	50.1
	EN-Basic	61.2	74.2	56.2	52.8	62.0	52.0	50.6	50.6	50.2	46.4	48.4	54.3
	Native-CoT	62.8	64.6	49.6	53.8	64.8	49.8	51.8	45.4	32.6	49.8	46.6	50.9
	EN-CoT	67.4	71.8	65.4	51.4	68.2	48.2	49.0	46.8	48.6	50.4	45.6	54.5
	XLT	65.4	72.6	66.2	57.2	70.0	47.0	49.2	50.8	50.2	50.6	46.6	56.0
	Trans-Google	77.8	80.4	75.2	75.0	76.4	66.6	67.6	74.0	71.8	68.8	63.2	**71.9**
	Trans-NLLB	73.0	75.6	69.6	74.4	73.2	67.4	62.4	73.8	66.2	68.0	51.2	68.2
Llama-2-70B-Chat	Native-Basic	61.6	81.6	49.4	49.4	55.4	50.6	46.8	49.8	41.0	46.4	44.6	51.5
	EN-Basic	74.6	79.4	58.0	53.6	63.2	48.8	50.2	49.4	50.4	49.0	51.0	55.3
	Native-CoT	65.8	78.0	52.2	51.8	54.8	49.2	49.2	50.2	40.0	43.2	36.2	50.5
	EN-CoT	80.4	88.0	73.6	65.4	77.8	53.0	50.0	56.0	48.0	49.8	50.6	61.2
	XLT	79.8	82.0	67.4	64.6	74.4	49.8	51.8	55.0	47.8	46.2	48.2	58.7
	Trans-Google	87.2	88.0	83.6	82.2	89.4	76.6	77.6	83.4	83.6	76.4	68.4	**80.9**
	Trans-NLLB	83.2	86.6	80.2	79.8	85.8	74.4	71.4	79.2	79.6	76.2	58.2	77.1

Table A.4: Accuracy scores across various languages on the XCOPA benchmark.

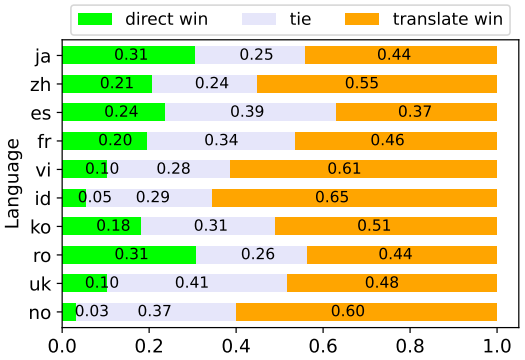
conditions: we heuristically filter queries using GPT-4o to ensure their validity, select 200 queries per language from the filtered set, and employ multiple judge models. Due to an insufficient number of available queries in other languages, we limit our evaluation to Japanese (ja), Chinese (zh), Spanish (es), French (fr), and Korean (ko). For the judging process, we use not only GPT-4o but also Claude-3.5-Sonnet and Gemini-Pro-1.5 to provide a more diverse assessment. The results are presented in Figure A.2. ChatGPT performs better when given direct prompts in languages such as Japanese and Chinese, whereas Llama-2-70B-Chat consistently achieves higher performance with translated prompts. These findings align with those discussed in Section 5.3.2.

Model	Prompt type	de	ru	fr	zh	es	vi	tr	sw	ar	el	th	bg	hi	ur	avg
ChatGPT	Native-Basic	59.0	58.8	60.2	54.0	60.2	49.2	51.6	51.0	50.6	58.0	39.6	54.8	42.8	40.4	52.2
	EN-Basic	68.6	58.2	67.4	62.2	68.4	63.0	65.6	65.2	62.4	64.6	56.4	65.4	55.8	59.0	**63.0**
	Native-CoT	59.4	54.2	58.0	51.8	58.6	47.6	53.0	50.8	51.2	54.6	37.2	54.4	40.2	37.4	50.6
	EN-CoT	62.6	56.4	61.4	57.4	65.8	57.6	58.0	54.0	53.4	59.0	51.0	59.8	48.6	45.0	56.4
	XLT	63.0	57.8	61.4	58.4	63.4	59.8	61.4	58.0	57.8	60.4	55.0	59.6	53.2	59.2	59.2
	Trans-Google	65.6	59.6	65.2	62.6	62.6	58.6	60.4	57.6	63.2	62.2	56.4	60.0	57.0	55.8	60.5
	Trans-NLLB	63.4	62.2	61.6	57.4	62.8	55.6	59.4	58.8	62.4	63.4	54.2	61.6	52.8	53.0	59.2
bloomz-7b1	Native-Basic	0.4	13.4	0.2	6.6	1.4	0.0	6.8	18.2	1.6	5.2	26.6	15.4	17.8	2.8	8.3
	EN-Basic	39.8	42.8	50.8	52.4	52.2	51.4	34.2	42.4	45.6	37.2	33.8	40.4	49.2	43.0	43.9
	Native-CoT	0.4	3.0	1.2	1.2	1.2	0.2	9.0	27.2	1.6	0.8	33.4	12.4	20.0	3.8	8.2
	EN-CoT	36.2	35.2	37.4	42.2	37.4	37.2	33.2	34.8	36.2	33.6	33.2	34.2	37.6	34.4	35.9
	XLT	38.2	34.4	35.0	34.0	35.0	36.0	37.4	35.4	34.6	35.6	35.0	36.6	33.8	34.0	35.4
	Trans-Google	45.0	43.4	44.2	44.0	45.2	44.8	43.8	44.0	44.0	44.6	44.4	44.8	43.4	44.4	**44.3**
	Trans-NLLB	45.6	43.0	44.0	44.0	45.4	42.4	43.6	43.4	44.6	44.6	43.2	44.8	42.8	42.0	43.8
Mistral-7B-Instruct	Native-Basic	50.4	55.6	59.2	46.0	59.0	33.4	38.8	33.0	34.2	34.2	39.2	46.6	37.0	33.2	42.8
	EN-Basic	56.4	54.6	59.8	54.0	56.8	51.4	46.8	37.6	45.8	49.4	47.0	54.4	46.4	41.8	50.2
	Native-CoT	50.0	55.0	58.4	47.6	54.6	35.8	38.2	32.2	37.6	35.4	40.0	52.0	36.8	33.8	43.4
	EN-CoT	55.0	52.2	58.0	52.4	57.0	50.4	48.0	38.0	48.6	51.2	45.8	54.2	46.8	42.0	50.0
	XLT	48.2	44.6	49.6	49.4	52.4	46.0	48.0	39.0	42.2	46.4	45.4	46.6	44.0	42.6	46.0
	Trans-Google	58.6	54.2	59.2	52.6	59.0	55.0	54.6	53.0	56.4	58.2	48.8	56.8	52.4	50.6	**55.0**
	Trans-NLLB	57.0	52.4	55.8	50.2	58.2	53.0	54.2	49.4	53.0	56.4	47.4	55.2	50.0	49.6	53.0
Llama-2-13b-Chat	Native-Basic	41.4	40.2	44.0	38.6	42.8	32.4	34.6	31.6	32.8	34.2	34.0	37.4	31.4	33.6	36.4
	EN-Basic	50.2	47.4	51.6	45.0	51.8	43.0	41.8	37.8	38.8	42.6	36.4	45.0	38.4	37.8	43.4
	Native-CoT	39.4	42.0	43.4	32.6	42.6	31.8	31.4	33.4	31.2	35.2	32.8	38.2	32.6	33.2	35.7
	EN-CoT	45.6	46.8	48.8	44.4	46.6	44.8	41.8	38.6	43.2	43.4	38.6	46.2	42.0	40.8	43.7
	XLT	59.6	55.8	56.4	54.0	59.8	55.6	48.2	37.8	49.4	49.0	44.4	52.0	48.4	49.2	**51.4**
	Trans-Google	50.4	44.2	45.4	44.6	46.0	46.0	47.6	42.8	48.4	48.2	43.4	45.4	45.4	47.4	46.1
	Trans-NLLB	48.6	46.6	47.0	43.2	44.6	43.6	49.0	43.2	44.0	46.0	41.6	48.6	45.6	43.4	45.4
Llama-2-70B-Chat	Native-Basic	44.0	42.0	45.4	42.6	45.6	38.4	38.4	32.6	35.0	37.6	33.0	41.8	34.8	34.8	39.0
	EN-Basic	53.6	54.6	57.0	49.6	55.6	46.0	42.8	32.4	50.2	46.2	38.6	52.4	37.6	34.8	46.5
	Native-CoT	40.4	42.2	45.4	38.4	41.4	38.4	36.6	32.8	35.2	37.4	32.6	41.0	33.2	36.2	37.9
	EN-CoT	53.6	52.8	56.4	50.4	56.8	46.0	40.6	33.4	44.6	47.8	38.2	48.2	37.6	36.6	45.9
	XLT	56.0	59.4	59.6	55.2	61.2	52.6	51.4	36.4	44.4	55.4	44.6	57.8	51.2	45.8	52.2
	Trans-Google	58.8	53.4	56.8	56.4	54.8	51.8	55.4	49.6	57.2	56.4	50.2	57.4	50.8	46.6	**54.0**
	Trans-NLLB	56.4	52.8	54.6	49.8	58.6	50.2	53.4	51.0	52.0	56.0	48.8	52.4	49.0	45.6	52.2

Table A.5: Accuracy scores across various languages on the XNLI benchmark.



(A) Win rate with ChatGPT (w/ cultural knowledge)



(B) Win rate with Llama-2-70B-Chat (w/ cultural knowledge)

Figure A.1: Win rate comparison for each language using ChatGPT and Llama-2-70B-Chat for the subsets of shareGPT with cultural knowledge.

Model	Prompt type	de	fr	zh	es	ja	ko	avg
ChatGPT	Native-Basic	62.0	53.6	46.6	46.6	49.0	40.6	49.7
	EN-Basic	67.6	68.0	58.8	71.4	55.8	50.4	62.0
	Native-CoT	61.8	55.0	48.6	48.8	44.0	43.4	50.3
	EN-CoT	67.6	64.0	61.2	70.0	55.8	51.2	61.6
	XLT	57.4	63.8	59.8	59.2	58.2	58.2	59.4
	Trans-Google	69.0	69.6	66.0	71.4	65.0	62.0	67.2
	Trans-NLLB	67.0	70.6	68.6	70.2	65.4	63.4	**67.5**
Mistral-7B-Instruct	Native-Basic	40.6	47.0	49.2	44.2	41.8	44.8	44.6
	EN-Basic	46.8	47.8	47.8	46.8	45.8	46.6	46.9
	Native-CoT	43.8	50.2	38.8	43.6	45.0	44.2	44.3
	EN-CoT	46.2	47.4	47.8	47.0	44.8	46.4	46.6
	XLT	47.4	49.6	47.6	46.6	48.2	47.2	47.8
	Trans-Google	51.2	49.8	54.0	49.6	52.4	52.0	51.5
	Trans-NLLB	50.6	52.8	52.4	50.8	54.2	53.6	**52.4**
Llama-2-13b-Chat	Native-Basic	50.8	57.2	54.0	58.0	55.8	48.2	54.0
	EN-Basic	60.2	61.0	58.6	59.8	58.2	56.8	**59.1**
	Native-CoT	50.4	58.8	59.0	55.8	56.8	47.8	54.8
	EN-CoT	59.2	55.8	58.6	59.2	56.4	56.6	57.6
	XLT	54.8	58.0	53.6	56.6	56.8	54.2	55.7
	Trans-Google	56.6	62.0	59.6	61.6	56.2	54.6	58.4
	Trans-NLLB	56.2	60.0	57.4	59.4	57.6	56.2	57.8
Llama-2-70B-Chat	Native-Basic	53.4	49.8	55.6	61.0	46.8	50.4	52.8
	EN-Basic	62.8	66.2	58.4	67.0	55.2	52.8	60.4
	Native-CoT	53.0	53.4	53.6	65.4	54.6	49.6	54.9
	EN-CoT	65.0	70.8	65.0	70.2	61.6	57.0	64.9
	XLT	57.0	61.6	57.6	57.2	49.4	51.6	55.7
	Trans-Google	70.6	70.6	68.0	72.2	65.6	64.2	68.5
	Trans-NLLB	69.8	73.4	69.4	71.2	68.8	62.4	**69.2**

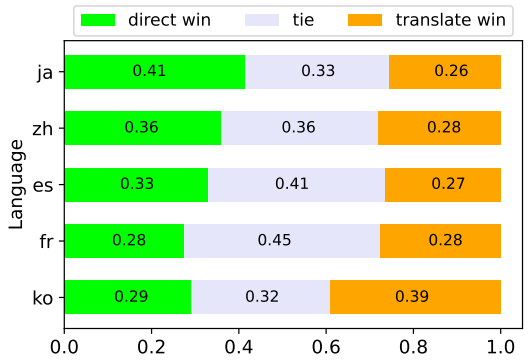
Table A.6: Accuracy scores across various languages on the PAWS-X benchmark.

Model	Prompt type	de	ru	fr	zh	es	ja	vi	tr	th	avg
ChatGPT	Native-Basic	44.1	30.5	46.4	31.4	40.2	20.2	33.0	39.2	33.6	35.4
	EN-Basic	36.9	30.5	43.3	28.9	44.1	43.5	34.7	32.7	34.0	36.5
	Native-CoT	43.6	22.2	46.1	30.0	38.3	33.9	34.1	38.3	31.9	35.4
	EN-CoT	44.6	37.4	49.7	38.5	48.0	52.4	32.6	42.0	40.5	42.9
	XLT	36.6	31.0	39.3	31.8	44.0	43.6	37.3	37.9	37.2	37.6
	ransg	42.0	39.2	42.7	48.6	40.8	46.4	37.8	44.2	52.3	**43.8**
	Trans-NLLB	39.2	34.6	26.7	31.6	29.1	45.3	41.2	41.2	45.9	37.2
bloomz-7b1	Native-Basic	0.6	3.0	7.6	12.1	11.2	7.5	7.6	0.0	20.9	7.8
	EN-Basic	7.5	3.7	12.3	21.4	12.2	12.3	13.3	2.1	11.0	10.6
	Native-CoT	0.2	0.9	5.9	8.6	8.3	6.0	6.7	0.0	22.2	6.5
	EN-CoT	4.0	3.0	11.4	17.9	13.9	8.7	11.1	1.7	12.2	9.3
	XLT	5.7	2.8	10.2	14.8	10.1	7.1	9.6	1.4	10.4	8.0
	Trans-Google	13.5	11.5	10.7	25.7	12.5	22.5	12.8	11.7	34.2	**17.2**
	Trans-NLLB	11.7	8.7	7.2	15.2	9.3	24.5	13.1	11.2	31.3	14.7
Mistral-7B-Instruct	Native-Basic	8.5	5.2	8.7	7.2	9.5	7.4	8.0	2.6	13.5	7.8
	EN-Basic	7.9	5.0	7.5	5.1	8.7	6.7	6.3	5.3	10.6	7.0
	Native-CoT	9.1	5.4	7.7	8.1	8.2	7.9	7.3	2.8	13.6	7.8
	EN-CoT	11.2	7.8	16.0	8.4	14.9	13.1	7.9	7.6	16.4	11.5
	XLT	9.7	7.2	10.4	8.4	10.4	10.5	9.2	6.6	14.2	9.6
	Trans-Google	14.6	13.8	14.9	17.7	17.0	22.5	13.4	15.1	24.4	**17.0**
	Trans-NLLB	13.3	12.7	10.5	14.9	11.8	24.1	13.8	13.5	25.2	15.5
Llama-2-13b-Chat	Native-Basic	15.0	13.6	31.3	20.6	29.7	13.8	21.2	5.8	13.4	18.3
	EN-Basic	28.5	11.6	28.7	13.9	27.2	21.0	15.3	15.6	20.0	20.2
	Native-CoT	14.6	10.4	29.1	13.3	23.6	10.5	23.8	5.6	10.1	15.7
	EN-CoT	28.2	12.6	31.1	11.9	28.9	15.3	15.4	18.3	16.3	19.8
	XLT	23.6	17.0	27.5	10.3	26.2	18.2	14.7	16.4	17.2	19.0
	Trans-Google	31.1	29.9	34.6	35.1	31.7	35.4	30.8	31.7	43.9	**33.8**
	Trans-NLLB	26.1	26.6	19.8	27.4	18.5	36.2	32.1	29.2	40.2	28.4
Llama-2-70B-Chat	Native-Basic	36.7	23.8	35.2	15.9	39.3	24.7	26.7	8.6	12.1	24.8
	EN-Basic	33.2	18.1	32.9	18.8	33.7	26.6	16.3	20.7	22.3	24.7
	Native-CoT	34.8	19.5	33.9	13.1	38.5	13.1	24.1	9.2	10.6	21.9
	EN-CoT	39.5	24.6	39.0	24.2	41.0	35.2	25.3	26.4	25.0	31.1
	XLT	29.8	22.4	29.6	18.0	31.3	29.5	25.0	27.3	26.1	26.6
	Trans-Google	37.3	34.0	37.1	43.5	35.4	48.0	35.8	38.3	47.9	**39.7**
	Trans-NLLB	16.7	16.4	11.9	18.5	14.9	26.8	19.7	21.8	27.6	19.4

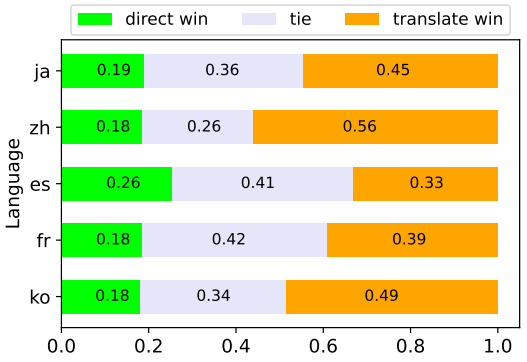
Table A.7: F1 scores across various languages on the MKQA benchmark.

Model	Prompt type	fr	zh	es	vi	tr	avg
ChatGPT	Native-Basic	29.2	39.3	26.9	34.4	29.9	31.9
	EN-Basic	28.9	38.8	27.8	37.9	30.0	**32.7**
	Native-CoT	28.8	38.5	26.1	34.0	27.9	31.1
	EN-CoT	25.4	35.1	26.0	33.5	28.6	29.7
	XLT	24.2	25.5	18.1	23.4	26.1	23.4
	Trans-Google	27.2	36.2	26.3	32.6	28.9	30.3
	Trans-NLLB	26.4	29.7	26.1	31.5	27.7	28.3
bloomz-7b1	Native-Basic	14.6	24.3	20.0	7.7	8.2	14.9
	EN-Basic	20.1	23.9	20.9	20.6	14.2	**19.9**
	Native-CoT	18.2	25.4	24.1	1.7	8.0	15.5
	EN-CoT	18.0	26.1	21.6	19.3	11.3	19.3
	XLT	12.2	19.9	19.3	14.5	5.3	14.2
	Trans-Google	10.0	14.2	12.1	9.0	10.7	11.2
	Trans-NLLB	10.5	8.6	12.5	9.7	11.5	10.6
Mistral-7B-Instruct	Native-Basic	23.0	34.0	22.3	25.8	24.4	25.9
	EN-Basic	20.9	16.5	21.5	28.0	21.1	21.6
	Native-CoT	19.7	33.6	22.1	26.4	21.1	24.6
	EN-CoT	20.6	12.1	19.9	22.2	18.8	18.7
	XLT	15.4	16.5	14.7	21.7	19.6	17.6
	Trans-Google	26.8	34.9	26.4	39.5	36.5	**32.8**
	Trans-NLLB	26.8	30.0	26.6	40.6	37.3	32.2
Llama-2-13b-Chat	Native-Basic	27.7	21.9	25.3	38.8	23.8	27.5
	EN-Basic	25.7	38.2	23.6	37.7	30.2	31.1
	Native-CoT	27.9	29.0	24.8	31.8	22.7	27.2
	EN-CoT	24.0	39.4	23.1	36.4	28.0	30.2
	XLT	24.2	17.7	22.4	23.6	18.1	21.2
	Trans-Google	28.0	42.9	27.9	41.6	38.0	**35.7**
	Trans-NLLB	27.5	37.5	26.9	41.6	37.3	34.2
Llama-2-70B-Chat	Native-Basic	28.8	34.5	27.3	29.7	26.8	29.4
	EN-Basic	29.0	31.8	24.3	35.7	35.3	31.2
	Native-CoT	25.3	29.5	26.7	22.4	25.2	25.8
	EN-CoT	27.0	35.2	22.1	34.8	32.0	30.2
	XLT	18.1	29.7	15.2	14.2	11.5	17.7
	Trans-Google	26.8	39.7	27.1	38.7	36.7	**33.8**
	Trans-NLLB	26.6	37.5	26.3	39.0	36.2	33.1

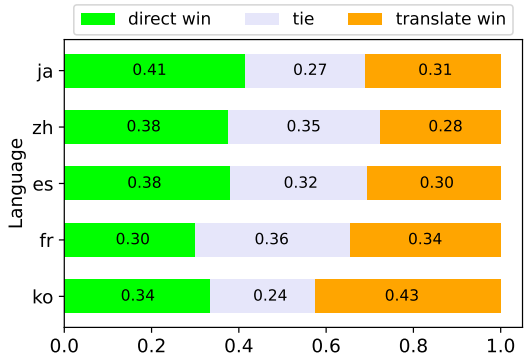
Table A.8: ROUGE-1 scores across various languages on the XL-sum benchmark.



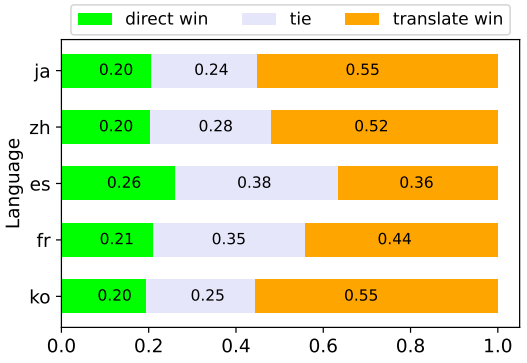
(A) ChatGPT judged by GPT-4o



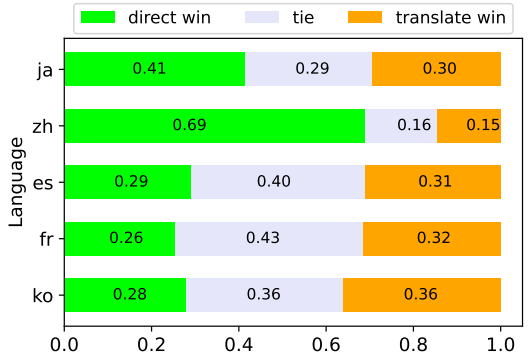
(B) Llama-2-70B-Chat judged by GPT-4o



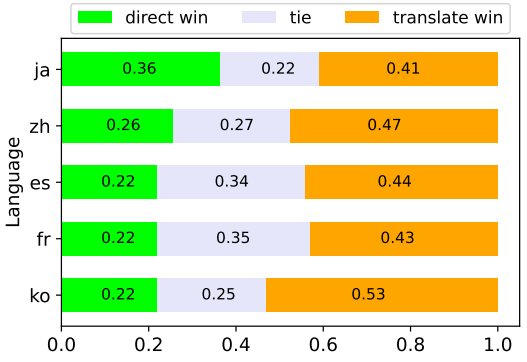
(C) ChatGPT judged by Claude-3.5-Sonnet



(D) Llama-2-70B-Chat judged by Claude-3.5-Sonnet



(E) ChatGPT judged by Gemini-1.5-Pro



(F) Llama-2-70B-Chat judged by Gemini-1.5-Pro

FIGURE A.2: Win rate comparison for five languages using ChatGPT and Llama-2-70B-Chat judged with three advanced LLMs.

List of Author's Awards, Patents, and Publications[1]

Conference Proceedings

- **Chaoqun Liu**, Wenxuan Zhang, Jiahao Ying, Mahani Aljunied, Anh Tuan Luu and Lidong Bing, "SeaExam and SeaBench: Benchmarking LLMs with Local Multilingual Questions in Southeast Asia," in *Findings of the North American Chapter of the Association for Computational Linguistics (NAACL), 2025.* [151]

- **Chaoqun Liu**, Wenxuan Zhang, Yiran Zhao, Anh Tuan Luu and Lidong Bing, "Is Translation All You Need? A Study on Solving Multilingual Tasks with Large Language Models," in *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2025.* [152]

- **Chaoqun Liu**, Qin Chao, Wenxuan Zhang, Xiaobao Wu, Boyang Li, Anh Tuan Luu and Lidong Bing, "Zero-to-Strong Generalization: Eliciting Strong Capabilities of Large Language Models Iteratively without Gold Labels," in *Proceedings of the International Conference on Computational Linguistics (COLING), 2025.* [153]

- **Chaoqun Liu**, Wenxuan Zhang, Guizhen Chen, Xiaobao Wu, Anh Tuan Luu, Chip Hong Chang and Lidong Bing, "Zero-Shot Text Classification via Self-Supervised Tuning," in *Findings of the Association for Computational Linguistics (ACL), 2023.* [154]

[1]The superscript * indicates joint first authors

- Yew Ken Chia, Liying Cheng, Hou Pong Chan, **Chaoqun Liu**, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria and Lidong Bing, "M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing.* [155]

- Guizhen Chen, Weiwen Xu, Hao Zhang, Hou Pong Chan, **Chaoqun Liu**, Lidong Bing, Deli Zhao, Anh Tuan Luu and Yu Rong, "FINEREASON: Evaluating and Improving LLMs' Deliberate Reasoning through Reflective Puzzle Solving," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), 2025.* [156]

- Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Zhiqiang Hu, Chenhui Shen, Yew Ken Chia, Xingxuan Li, Jianyu Wang, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, **Chaoqun Liu**, Hang Zhang and Lidong Bing, "SeaLLMs - Large Language Models for Southeast Asia," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. [157]

- Wenxuan Zhang, Hou Pong Chan, Yiran Zhao, Mahani Aljunied, Jianyu Wang, **Chaoqun Liu**, Yue Deng, Zhiqiang Hu, Weiwen Xu, Yew Ken Chia, Xin Li and Lidong Bing, "SeaLLMs 3: Open Foundation and Chat Multilingual Large Language Models for Southeast Asian Languages," in *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025. [158]

- Xiaobao Wu, Fengjun Pan, Thong Nguyen, Yichao Feng, **Chaoqun Liu**, Cong-Duy Nguyen and Anh Tuan Luu, "On the affinity, rationality, and diversity of hierarchical topic modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence, 2024.* [159]

- Xiaobao Wu, Xinshuai Dong, Thong Nguyen, **Chaoqun Liu**, Liangming Pan and Anh Tuan Luu, "InfoCTM: A Mutual Information Maximization Perspective of Cross-lingual Topic Modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence, 2023.* [160]

Preprints

- **Chaoqun Liu**, Mahani Aljunied, Guizhen Chen, Hou Pong Chan, Weiwen Xu, Yu Rong and Wenxuan Zhang, "SeaLLMs-Audio: Large Audio-Language Models for Southeast Asia," *arXiv preprint arXiv:2511.01670*, 2025. [161]

- LASA Team, Weiwen Xu, Hou Pong Chan, Long Li, Mahani Aljunied, Ruifeng Yuan, Jianyu Wang, Chenghao Xiao, Guizhen Chen, **Chaoqun Liu**, Zhaodonghui Li, Yu Sun, Junao Shen, Chaojun Wang, Jie Tan, Deli Zhao, Tingyang Xu, Hao Zhang and Yu Rong, "Lingshu: A Generalist Foundation Model for Unified Multimodal Medical Understanding and Reasoning," *arXiv preprint arXiv:2506.07044*, 2025. [162]

- Yiran Zhao, **Chaoqun Liu**, Yue Deng, Jiahao Ying, Mahani Aljunied, Zhaodonghui Li, Lidong Bing, Hou Pong Chan, Yu Rong, Deli Zhao and Wenxuan Zhang, "Babel: Open Multilingual Large Language Models Serving Over 90% of Global Speakers," *arXiv preprint arXiv:2503.00865*, 2025. [163]

Bibliography

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`. xv, 8, 17

[2] Mozes van de Kar, Mengzhou Xia, Danqi Chen, and Mikel Artetxe. Don't prompt, search! mining-based zero-shot learning with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022. URL `https://doi.org/10.48550/arXiv.2210.14803`. xv, 1, 8, 9, 17, 18, 26, 28

[3] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022. URL `https://openreview.net/forum?id=gEZrGCozdqR`. xv, 1, 8, 9, 17, 18

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. URL `https://doi.org/10.18653/v1/n19-1423`. xv, 1, 10, 11, 17, 18, 22, 23, 32

[5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer.

BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.703. URL https://doi.org/10.18653/v1/2020.acl-main.703. xv, 10, 11

[6] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html. xv, 11

[7] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. Structbert: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BJgQ4lSFPH. xv, 11, 12

[8] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision, December 2023. URL http://arxiv.org/abs/2312.09390. arXiv:2312.09390 [cs]. xv, xvi, 2, 12, 13, 39, 40

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. URL http://arxiv.org/abs/2005.14165. arXiv:2005.14165 [cs]. xv, 2, 13, 14, 61

[10] Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. Not All Languages Are Created Equal in LLMs: Improving Multilingual Capability by Cross-Lingual-Thought Prompting, May 2023. URL http://arxiv.org/abs/2305.07004. arXiv:2305.07004 [cs]. xv, 2, 14, 61, 62, 64, 67, 68, 81

[11] Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3007. URL https://www.aclweb.org/anthology/P19-3007. xv, 34

[12] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models, March 2023. URL `http://arxiv.org/abs/2203.11171`. arXiv:2203.11171 [cs]. xvi, 41, 42

[13] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language Models are Multilingual Chain-of-Thought Reasoners, October 2022. URL `http://arxiv.org/abs/2210.03057`. arXiv:2210.03057 [cs]. xvi, 3, 14, 15, 61, 62, 64, 72, 81

[14] Wenxuan Zhang, Sharifah Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. M3Exam: A Multilingual, Multimodal, Multilevel Benchmark for Examining Large Language Models, June 2023. URL `http://arxiv.org/abs/2306.05179`. arXiv:2306.05179 [cs]. xvi, xx, 62, 75, 76, 81, 85, 86

[15] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, January 2023. URL `http://arxiv.org/abs/2201.11903`. arXiv:2201.11903 [cs]. xx, 47, 48

[16] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset, November 2021. URL `http://arxiv.org/abs/2103.03874`. arXiv:2103.03874 [cs]. xx, 3, 81, 85

[17] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. 1, 10, 17

[18] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=9Vrb9D0WI4`. 1, 8, 17, 18

[19] Ping Yang, Junjie Wang, Ruyi Gan, Xinyu Zhu, Lin Zhang, Ziwei Wu, Xinyu Gao, Jiaxing Zhang, and Tetsuya Sakai. Zero-shot learners for natural language understanding via a unified multiple choice perspective. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022. URL `https://doi.org/10.48550/arXiv.2210.08590`. 1, 9, 17, 18, 26, 28, 106

[20] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single QA system. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1896–1907. Association for Computational Linguistics, 2020. URL `https://doi.org/10.18653/v1/2020.findings-emnlp.171`. 1, 8, 18

[21] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=H1eA7AEtvS`. 1, 10, 11, 18, 23, 28

[22] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL `https://aclanthology.org/2022.emnlp-main.759`. 2, 13, 40

[23] Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. Ground-Truth Labels Matter: A Deeper Look into Input-Label Demonstrations, October 2022. URL `http://arxiv.org/abs/2205.12685`. arXiv:2205.12685 [cs]. 2, 13, 40, 44, 45, 46

[24] Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.153. URL `https://aclanthology.org/2023.acl-long.153`. 2, 13, 40, 47, 55

[25] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus,

Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling Language Modeling with Pathways, October 2022. URL `http://arxiv.org/abs/2204.02311`. arXiv:2204.02311 [cs]. 2, 61

[26] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, et al. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model, June 2023. URL `http://arxiv.org/abs/2211.05100`. arXiv:2211.05100 [cs]. 2, 61

[27] Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. MEGA: Multilingual Evaluation of Generative AI. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.258. URL `https://aclanthology.org/2023.emnlp-main.258`. 2, 15, 61, 81

[28] Viet Lai, Nghia Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Nguyen. ChatGPT Beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13171–13189, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.878. URL `https://aclanthology.org/2023.findings-emnlp.878`. 15, 68, 81

[29] Xiang Zhang, Senyu Li, Bradley Hauer, Ning Shi, and Grzegorz Kondrak. Don't Trust ChatGPT when your Question is not in English: A Study of Multilingual Abilities and Types of LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7915–7927, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.491. URL `https://aclanthology.org/2023.emnlp-main.491`. 2, 61

[30] Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. XCOPA: A Multilingual Dataset for Causal Commonsense Reasoning. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online, November 2020. Association for Computational Linguistics. doi: 10.

18653/v1/2020.emnlp-main.185. URL https://aclanthology.org/2020.emnlp-main.185. 2, 13, 61, 64, 72

[31] Nikita Moghe, Tom Sherborne, Mark Steedman, and Alexandra Birch. Extrinsic Evaluation of Machine Translation Metrics. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13060–13078, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.730. URL https://aclanthology.org/2023.acl-long.730. 13, 61

[32] Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. Cross-lingual Prompting: Improving Zero-shot Chain-of-Thought Reasoning across Languages. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.163. URL https://aclanthology.org/2023.emnlp-main.163. 2, 14, 61

[33] OpenAI. GPT-4 Technical Report, March 2023. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs]. 3, 61, 81

[34] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating Cross-lingual Sentence Representations. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1269. URL https://aclanthology.org/D18-1269. 3, 13, 61, 64, 72, 81

[35] Chaoqun Liu, Wenxuan Zhang, Yiran Zhao, Anh Tuan Luu, and Lidong Bing. Is Translation All You Need? A Study on Solving Multilingual Tasks with Large Language Models, March 2024. URL http://arxiv.org/abs/2403.10258. arXiv:2403.10258 [cs]. 3, 82

[36] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, December 2023. URL http://arxiv.org/abs/2306.05685. arXiv:2306.05685 [cs]. 3, 15, 73, 81, 87, 93, 100, 103

[37] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng

Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022. URL `https://doi.org/10.48550/arXiv.2204.02311`. 8, 17

[38] Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, June 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.naacl-main.185`. 8, 17, 28

[39] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.20. URL `https://doi.org/10.18653/v1/2021.eacl-main.20`. 8, 17

[40] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. In *Advances in Neural Information Processing Systems*, 2022. 8, 17

[41] Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2856–2878. Association for Computational Linguistics, 2021. URL `https://doi.org/10.18653/v1/2021.findings-emnlp.244`. 9, 18

[42] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3912–3921, 2019. URL `https://doi.org/10.18653/v1/D19-1404`. 9, 18, 28

[43] Hantian Ding, Jinrui Yang, Yuqian Deng, Hongming Zhang, and Dan Roth. Towards open-domain topic classification. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, 2022. 9, 18, 28

[44] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021. URL `https://doi.org/10.1109/TKDE.2021.3090866`. 10

[45] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 10

[46] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL `http://arxiv.org/abs/1907.11692`. 10, 23, 28

[47] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77, 2020. URL `https://doi.org/10.1162/tacl_a_00300`. 10

[48] Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. COCO-LM: correcting and contrasting text sequences for language model pretraining. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 23102–23114, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/c2c2a04512b35d13102459f8784f1a2d-Abstract.html`. 12

[49] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating Training Data with Language Models: Towards Zero-Shot Language Understanding, October 2022. URL `http://arxiv.org/abs/2202.04538`. arXiv:2202.04538 [cs]. 12

[50] Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. ZeroGen: Efficient Zero-shot Learning via Dataset Generation, October 2022. URL `http://arxiv.org/abs/2202.07922`. arXiv:2202.07922 [cs]. 12

[51] Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. Learning the structure of generative models without labeled data. In *International Conference on Machine Learning*, pages 273–282. PMLR, 2017. 12

[52] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019. 12

[53] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

[54] Chaoqun Liu, Wenxuan Zhang, Guizhen Chen, Xiaobao Wu, Anh Tuan Luu, Chip Hong Chang, and Lidong Bing. Zero-shot text classification via self-supervised tuning. pages 1743–1761, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.findings-acl.110. URL `2023.findings-acl.110`.

[55] Fengjun Pan, Xiaobao Wu, Zongrui Li, and Anh Tuan Luu. Are LLMs good zero-shot fallacy classifiers? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14338–14364, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.794. URL `https://aclanthology.org/2024.emnlp-main.794`.

[56] Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. AKEW: Assessing knowledge editing in the wild. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15118–15133, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.emnlp-main.843`. 12

[57] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022. 12

[58] Xiaonan Li and Xipeng Qiu. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6354–6374, 2023.

[59] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping Reasoning With Reasoning, May 2022. URL `http://arxiv.org/abs/2203.14465`. arXiv:2203.14465 [cs]. 12

[60] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021. 13

[61] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR, 2021.

[62] Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*, 2022. 13

[63] Liang Wang, Nan Yang, and Furu Wei. Learning to Retrieve In-Context Examples for Large Language Models. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.105. 13

[64] Mikel Artetxe, Vedanuj Goswami, Shruti Bhosale, Angela Fan, and Luke Zettlemoyer. Revisiting Machine Translation for Cross-lingual Classification. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6489–6499, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.399. URL https://aclanthology.org/2023.emnlp-main.399. 13, 61

[65] Kunxun Qi, Hai Wan, Jianfeng Du, and Haolan Chen. Enhancing Cross-lingual Natural Language Inference by Prompt-learning from Cross-lingual Templates. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1910–1923, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.134. URL https://aclanthology.org/2022.acl-long.134.

[66] Lianzhe Huang, Shuming Ma, Dongdong Zhang, Furu Wei, and Houfeng Wang. Zero-shot Cross-lingual Transfer of Prompt-based Tuning with a Unified Multilingual Prompt, October 2022. URL http://arxiv.org/abs/2202.11451. arXiv:2202.11451 [cs]. 13

[67] Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. Do Multilingual Language Models Think Better in English?, August 2023. URL http://arxiv.org/abs/2308.01223. arXiv:2308.01223 [cs]. 14, 61, 62, 70

[68] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is ChatGPT a General-Purpose Natural Language Processing Task Solver?, February 2023. URL http://arxiv.org/abs/2302.06476. arXiv:2302.06476 [cs]. 15

[69] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity, February 2023. URL http://arxiv.org/abs/2302.04023. arXiv:2302.04023 [cs]. 15

[70] Yi Fung, Ruining Zhao, Jae Doo, Chenkai Sun, and Heng Ji. Massively Multi-Cultural Knowledge Acquisition & LM Benchmarking, February 2024. URL `http://arxiv.org/abs/2402.09369`. arXiv:2402.09369 [cs]. 15

[71] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 5 2023. 15

[72] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024. 15

[73] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline, June 2024. URL `http://arxiv.org/abs/2406.11939`. arXiv:2406.11939 [cs]. 15

[74] Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. WildBench: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild, June 2024. URL `http://arxiv.org/abs/2406.04770`. arXiv:2406.04770 [cs]. 15

[75] Rishav Hada, Varun Gumma, Adrian Wynter, Harshita Diddee, Mohamed Ahmed, Monojit Choudhury, Kalika Bali, and Sunayana Sitaram. Are large language model-based evaluators the solution to scaling up multilingual evaluation? In Yvette Graham and Matthew Purver, editors, *FINDINGS:2024:eacl*, pages 1051–1070, St. Julian's, Malta, March 2024. acl. URL `https://aclanthology.org/2024.findings-eacl.71`. 15

[76] Bin Wang, Zhengyuan Liu, Xin Huang, Fangkai Jiao, Yang Ding, Ai Ti Aw, and Nancy F. Chen. SeaEval for Multilingual Foundation Models: From Cross-Lingual Alignment to Cultural Reasoning, September 2023. URL `http://arxiv.org/abs/2309.04766`. arXiv:2309.04766 [cs]. 16

[77] Holy Lovenia, Rahmad Mahendra, Salsabil Maulana Akbar, et al. SEACrowd: A Multilingual Multimodal Data Hub and Benchmark Suite for Southeast Asian Languages, June 2024. URL `http://arxiv.org/abs/2406.10118`. arXiv:2406.10118 [cs]. 16

[78] Wei Qi Leong, Jian Gang Ngui, Yosephine Susanto, Hamsawardhini Rengarajan, Kengatharaiyer Sarveswaran, and William Chandra Tjhi. BHASA: A Holistic Southeast Asian Linguistic and Cultural Evaluation Suite for Large Language Models. 16

[79] Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. *CoRR*, abs/2202.07922, 2022. URL `https://arxiv.org/abs/2202.07922`. 17

[80] Jiangshu Du, Wenpeng Yin, Congying Xia, and Philip S. Yu. Learning to select from multiple options. In *Proceedings of the 2023 AAAI*, 2023. URL https://doi.org/10.48550/arXiv.2212.00301. 18

[81] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.552. URL https://doi.org/10.18653/v1/2021.emnlp-main.552. 21

[82] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, 2019. 24

[83] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html. 26

[84] Ken Lang. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 331–339. Morgan Kaufmann, 1995. URL https://doi.org/10.1016/b978-1-55860-377-6.50048-7. 26

[85] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL, 2013. URL https://aclanthology.org/D13-1170/. 26

[86] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. Association for Computational Linguistics, 2011. URL https://aclanthology.org/P11-1015/. 26

[87] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer, editors, *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 115–124. The Association for Computer Linguistics, 2005. URL `https://aclanthology.org/P05-1015/`. 26

[88] Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. Zero-shot text classification with self-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022. URL `https://arxiv.org/abs/2210.17541`. 26, 28

[89] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Sasko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In Heike Adel and Shuming Shi, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2021, Online and Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 175–184. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-demo.21. URL `https://doi.org/10.18653/v1/2021.emnlp-demo.21`. 26

[90] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.295. URL `https://doi.org/10.18653/v1/2021.acl-long.295`. 26

[91] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. 2018. 39

[92] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. URL `http://arxiv.org/abs/1810.04805`. arXiv:1810.04805 [cs].

[93] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language

Models Are Zero-Shot Learners, February 2022. URL http://arxiv.org/abs/2109.01652. arXiv:2109.01652 [cs].

[94] Victor Sanh, Albert Webson, Colin Raffel, et al. Multitask Prompted Training Enables Zero-Shot Task Generalization, March 2022. URL http://arxiv.org/abs/2110.08207. arXiv:2110.08207 [cs]. 39

[95] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 Herd of Models, August 2024. URL http://arxiv.org/abs/2407.21783. arXiv:2407.21783 [cs]. 44, 58, 81, 93

[96] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, October 2023. URL http://arxiv.org/abs/2310.06825. arXiv:2310.06825 [cs]. 44, 61, 66, 93

[97] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context LLMs Struggle with Long In-context Learning, April 2024. URL http://arxiv.org/abs/2404.02060. arXiv:2404.02060 [cs]. 45

[98] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. Goemotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*, 2020. 45

[99] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient Intent Detection with Dual Sentence Encoders. In Tsung-Hsien Wen, Asli Celikyilmaz, Zhou Yu, Alexandros Papangelis, Mihail Eric, Anuj Kumar, Iñigo Casanueva, and Rushin Shah, editors, *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5. URL https://aclanthology.org/2020.nlp4convai-1.5. 45

[100] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 45, 46

[101] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168. 45, 46

[102] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://aclanthology.org/D13-1170`. 45, 59

[103] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3540334270. doi: 10.1007/11736790_9. URL `https://doi.org/10.1007/11736790_9`. 45

[104] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL `https://aclanthology.org/I05-5002`. 45

[105] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press, 2012. ISBN 9781577355601. 45

[106] Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, 23(2):107–124, Jul. 2019. doi: 10.18148/sub/2019.v23i2.601. URL `https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/601`. 45

[107] Ellen M. Voorhees and Dawn M. Tice. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132263. doi: 10.1145/345508.345577. URL `https://doi.org/10.1145/345508.345577`. 45

[108] Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.*, 65(4):782–796, apr 2014. ISSN 2330-1635. doi: 10.1002/asi.23062. URL `https://doi.org/10.1002/asi.23062`. 45

[109] Emily Sheng and David Uthus. Investigating societal biases in a poetry composition system. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 93–106, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. URL `https://aclanthology.org/2020.gebnlp-1.9`. 45

[110] Clara H. McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. Effective transfer learning for identifying similar questions: Matching user questions to covid-19 faqs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3458–3465, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3412861. URL https://doi.org/10.1145/3394486.3412861. 45

[111] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf. 45

[112] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5102. URL https://aclanthology.org/W18-5102. 45

[113] Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. Ethos: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, Jan 2022. ISSN 2198-6053. doi: 10.1007/s40747-021-00608-2. URL http://dx.doi.org/10.1007/s40747-021-00608-2. 45

[114] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.148. URL https://aclanthology.org/2020.findings-emnlp.148. 45

[115] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners, January 2023. URL http://arxiv.org/abs/2205.11916. arXiv:2205.11916 [cs]. 47

[116] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs]. 56

[117] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne

Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of Experts, January 2024. URL `http://arxiv.org/abs/2401.04088`. arXiv:2401.04088 [cs]. 58

[118] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021. URL `http://arxiv.org/abs/2110.14168`. arXiv:2110.14168 [cs]. 59

[119] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual Generalization through Multitask Finetuning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.891. URL `https://aclanthology.org/2023.acl-long.891`. 61, 66

[120] Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, Hang Zhang, and Lidong Bing. SeaLLMs – Large Language Models for Southeast Asia, December 2023. URL `http://arxiv.org/abs/2312.00738`. arXiv:2312.00738 [cs]. 61, 81

[121] Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. Few-shot Learning with Multilingual Generative Language Models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.616. URL `https://aclanthology.org/2022.emnlp-main.616`. 61

[122] Patrick Bareiß, Roman Klinger, and Jeremy Barnes. English Prompts are Better for NLI-based Zero-Shot Emotion Classification than Target-Language Prompts. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 1318–1326, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701726. doi: 10.1145/3589335.3651902. URL `https://doi.org/10.1145/3589335.3651902`. 61

[123] Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1382. URL `https://aclanthology.org/D19-1382`. 64, 65, 72

[124] Shayne Longpre, Yi Lu, and Joachim Daiber. MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406, 2021. doi: 10.1162/tacl_a_00433. URL `https://aclanthology.org/2021.tacl-1.82`. Place: Cambridge, MA Publisher: MIT Press. 64, 65

[125] Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.413. URL `https://aclanthology.org/2021.findings-acl.413`. 64, 65, 72, 77

[126] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, July 2023. URL `http://arxiv.org/abs/2307.09288`. arXiv:2307.09288 [cs]. 66

[127] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration, 2023. 66

[128] NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam,

Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No Language Left Behind: Scaling Human-Centered Machine Translation, August 2022. URL `http://arxiv.org/abs/2207.04672`. arXiv:2207.04672 [cs]. 67

[129] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://aclanthology.org/P02-1040`. 70

[130] Matt Post. A Call for Clarity in Reporting BLEU Scores. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL `https://aclanthology.org/W18-6319`. 70

[131] Fred Philippy, Siwen Guo, and Shohreh Haddadan. Identifying the Correlation Between Language Distance and Cross-Lingual Transfer in a Multilingual Representation Space. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 22–29, 2023. doi: 10.18653/v1/2023.sigtyp-1.3. URL `http://arxiv.org/abs/2305.02151`. arXiv:2305.02151 [cs]. 72

[132] Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 8–14, 2017. 72

[133] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang,

Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 75

[134] 01 AI, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open Foundation Models by 01.AI, March 2024. URL `http://arxiv.org/abs/2403.04652`. arXiv:2403.04652 [cs] version: 1. 75

[135] Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. How do Large Language Models Handle Multilingualism?, February 2024. URL `http://arxiv.org/abs/2402.18815`. arXiv:2402.18815 [cs]. 77

[136] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, et al. Gemma 2: Improving Open Language Models at a Practical Size, August 2024. URL `http://arxiv.org/abs/2408.00118`. arXiv:2408.00118 [cs]. 81, 93

[137] Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S. Yu. Multilingual Large Language Model: A Survey of Resources, Taxonomy and Frontiers, April 2024. URL `http://arxiv.org/abs/2404.04925`. arXiv:2404.04925 [cs]. 81

[138] Kaiyu Huang, Fengran Mo, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, Jian-Yun Nie, and Yang Liu. A Survey on Large Language Models with Multilingualism: Recent Advances and New Frontiers, May 2024. URL `http://arxiv.org/abs/2405.10936`. arXiv:2405.10936 [cs].

[139] Longxu Dou, Qian Liu, Guangtao Zeng, Jia Guo, Jiahui Zhou, Wei Lu, and Min Lin. Sailor: Open Language Models for South-East Asia, April 2024. URL `http://arxiv.org/abs/2404.03608`. arXiv:2404.03608 [cs]. 93

[140] Wenxuan Zhang, Hou Pong Chan, Yiran Zhao, Mahani Aljunied, Jianyu Wang, Chaoqun Liu, Yue Deng, Zhiqiang Hu, Weiwen Xu, Yew Ken Chia, Xin Li, and Lidong Bing. SeaLLMs 3: Open Foundation and Chat Multilingual Large Language Models for Southeast Asian Languages, July 2024. URL `http://arxiv.org/abs/2407.19672`. arXiv:2407.19672 [cs]. 81, 93

[141] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset, September 2023. URL `http://arxiv.org/abs/2309.11998`. arXiv:2309.11998 [cs]. 89

[142] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=Bl8u7ZRlbM`. 89

[143] Yuntian Deng, Wenting Zhao, Jack Hessel, Xiang Ren, Claire Cardie, and Yejin Choi. Wildvis: Open source visualizer for million-scale chat logs in the wild, 2024. URL `https://arxiv.org/abs/2409.03753`. 89

[144] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024. 90

[145] An Yang, Baosong Yang, Binyuan Hui, et al. Qwen2 Technical Report, July 2024. URL `http://arxiv.org/abs/2407.10671`. arXiv:2407.10671 [cs]. 93

[146] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools, July 2024. URL `http://arxiv.org/abs/2406.12793`. arXiv:2406.12793 [cs]. 93

[147] Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. Aya 23: Open Weight Releases to Further Multilingual Progress, May 2024. URL `http://arxiv.org/abs/2405.15032`. arXiv:2405.15032 [cs]. 93

[148] AI Singapore. Sea-lion (southeast asian languages in one network): A family of large language models for southeast asia. `https://github.com/aisingapore/sealion`, 2024. 93

[149] Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. Benchmarking foundation models with language-model-as-an-examiner, 2023. URL `https://arxiv.org/abs/2306.04181`. 93, 101

[150] Jiahao Ying, Yixin Cao, Yushi Bai, Qianru Sun, Bo Wang, Wei Tang, Zhaojun Ding, Yizhe Yang, Xuanjing Huang, and Shuicheng Yan. Automating dataset updates towards reliable and timely evaluation of large language models, 2024. URL `https://arxiv.org/abs/2402.11894`. 93, 101

[151] Chaoqun Liu, Wenxuan Zhang, Jiahao Ying, Mahani Aljunied, Anh Tuan Luu, and Lidong Bing. SeaExam and SeaBench: Benchmarking LLMs with Local Multilingual Questions in Southeast Asia. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6119–6136, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.341. URL `https://aclanthology.org/2025.findings-naacl.341/`. 119

[152] Chaoqun Liu, Wenxuan Zhang, Yiran Zhao, Anh Tuan Luu, and Lidong Bing. Is Translation All You Need? A Study on Solving Multilingual Tasks with Large Language Models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9594–9614, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.485. URL https://aclanthology.org/2025.naacl-long.485/. 119

[153] Chaoqun Liu, Qin Chao, Wenxuan Zhang, Xiaobao Wu, Boyang Li, Anh Tuan Luu, and Lidong Bing. Zero-to-Strong Generalization: Eliciting Strong Capabilities of Large Language Models Iteratively without Gold Labels. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3716–3731, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.251/. 119

[154] Chaoqun Liu, Wenxuan Zhang, Guizhen Chen, Xiaobao Wu, Anh Tuan Luu, Chip Hong Chang, and Lidong Bing. Zero-Shot Text Classification via Self-Supervised Tuning, May 2023. URL http://arxiv.org/abs/2305.11442. arXiv:2305.11442 [cs]. 119

[155] Yew Ken Chia, Liying Cheng, Hou Pong Chan, Maojia Song, Chaoqun Liu, Mahani Aljunied, Soujanya Poria, and Lidong Bing. M-LongDoc: A Benchmark For Multimodal Super-Long Document Understanding And A Retrieval-Aware Tuning Framework. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9233–9250, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.469. URL https://aclanthology.org/2025.emnlp-main.469/. 120

[156] Guizhen Chen, Weiwen Xu, Hao Zhang, Hou Pong Chan, Chaoqun Liu, Lidong Bing, Deli Zhao, Anh Tuan Luu, and Yu Rong. FINEREASON: Evaluating and Improving LLMs' Deliberate Reasoning through Reflective Puzzle Solving, February 2025. URL http://arxiv.org/abs/2502.20238. arXiv:2502.20238 [cs]. 120

[157] Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Zhiqiang Hu, Chenhui Shen, Yew Ken Chia, Xingxuan Li, Jianyu Wang, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, Hang Zhang, and Lidong Bing. SeaLLMs - Large Language Models for Southeast Asia. In Yixin Cao, Yang Feng, and Deyi Xiong, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 294–304, Bangkok, Thailand, August

2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.28. URL https://aclanthology.org/2024.acl-demos.28/. 120

[158] Wenxuan Zhang, Hou Pong Chan, Yiran Zhao, Mahani Aljunied, Jianyu Wang, Chaoqun Liu, Yue Deng, Zhiqiang Hu, Weiwen Xu, Yew Ken Chia, Xin Li, and Lidong Bing. SeaLLMs 3: Open Foundation and Chat Multilingual Large Language Models for Southeast Asian Languages. In Nouha Dziri, Sean (Xiang) Ren, and Shizhe Diao, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 96–105, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-191-9. doi: 10.18653/v1/2025.naacl-demo.10. URL https://aclanthology.org/2025.naacl-demo.10/. 120

[159] Xiaobao Wu, Fengjun Pan, Thong Nguyen, Yichao Feng, Chaoqun Liu, Cong-Duy Nguyen, and Anh Tuan Luu. On the affinity, rationality, and diversity of hierarchical topic modeling. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, volume 38 of *AAAI'24/IAAI'24/EAAI'24*, pages 19261–19269. AAAI Press, February 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29895. URL https://doi.org/10.1609/aaai.v38i17.29895. 120

[160] Xiaobao Wu, Xinshuai Dong, Thong Nguyen, Chaoqun Liu, Liang-Ming Pan, and Anh Tuan Luu. InfoCTM: a mutual information maximization perspective of cross-lingual topic modeling. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, volume 37 of *AAAI'23/IAAI'23/EAAI'23*, pages 13763–13771. AAAI Press, February 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i11.26612. URL https://doi.org/10.1609/aaai.v37i11.26612. 120

[161] Chaoqun Liu, Mahani Aljunied, Guizhen Chen, Hou Pong Chan, Weiwen Xu, Yu Rong, and Wenxuan Zhang. SeaLLMs-Audio: Large Audio-Language Models for Southeast Asia, November 2025. URL http://arxiv.org/abs/2511.01670. arXiv:2511.01670 [cs]. 121

[162] LASA Team, Weiwen Xu, Hou Pong Chan, Long Li, Mahani Aljunied, Ruifeng Yuan, Jianyu Wang, Chenghao Xiao, Guizhen Chen, Chaoqun Liu, Zhaodonghui Li, Yu Sun, Junao Shen, Chaojun Wang, Jie Tan, Deli Zhao, Tingyang Xu, Hao Zhang, and Yu Rong. Lingshu: A Generalist Foundation Model for Unified Multimodal Medical Understanding and Reasoning, June 2025. URL http://arxiv.org/abs/2506.07044. arXiv:2506.07044 [cs]. 121

[163] Yiran Zhao, Chaoqun Liu, Yue Deng, Jiahao Ying, Mahani Aljunied, Zhaodonghui Li, Lidong Bing, Hou Pong Chan, Yu Rong, Deli Zhao, and Wenxuan Zhang. Babel: Open Multilingual Large Language Models Serving Over 90% of Global Speakers, March 2025. URL `http://arxiv.org/abs/2503.00865`. arXiv:2503.00865 [cs]. 121