UltraLink: An Open-Source Knowledge-Enhanced Multilingual Supervised Fine-tuning Dataset

Anonymous ACL submission

Abstract

Open-source large language models (LLMs) have gained significant strength across diverse fields. Nevertheless, the majority of studies primarily concentrate on English, with only limited exploration into the realm of multilingual abilities. In this work, we therefore construct an open-source multilingual supervised fine-tuning dataset. Different from previous works that simply translate English instructions, we consider both the language-specific and language-agnostic abilities of LLMs. Firstly, we introduce a knowledge-grounded data augmentation approach to elicit more language-specific knowledge of LLMs, improving their ability to serve users from different countries. Moreover, we find modern LLMs possess strong cross-lingual transfer capabilities, thus repeatedly learning identical content in various languages is not necessary. Consequently, we can substantially prune the language-agnostic supervised fine-tuning (SFT) data without any performance degradation, making multilingual SFT more efficient. The resulting UltraLink dataset comprises approximately 1 million samples across five languages (i.e., En, Zh, Ru, Fr, Es), and the proposed data construction method can be easily extended to other languages. UltraLink-LM, which is trained on UltraLink, outperforms several representative baselines across many tasks.

1 Introduction

Thanks to the collaborative efforts of the active large language models (LLMs) community, open-source LLMs are becoming increasingly powerful (Touvron et al., 2023a,b; Jiang et al., 2023), even outperforming some representative closed-source counterparts (OpenAI, 2023; Anil et al., 2023) in some specific tasks (Wei et al., 2023b). These accomplishments are closely related to the contribution of open-source supervised fine-tuning (SFT) data (Ding et al., 2023; Anand et al., 2023; Peng et al., 2023; Wang et al., 2023; Kim et al., 2023; Xu et al., 2023), which plays a pivotal role in eliciting the instruction-following ability of LLMs and aligning the model behavior with human preferences. Nevertheless, the focus of existing works is primarily on the construction of English SFT data, resulting in a comparatively limited availability of multilingual SFT resources.

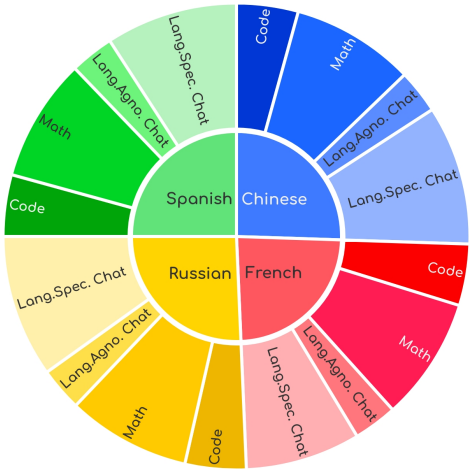

Figure 1: To equip large language models with not only language-specific knowledge but also language-agnostic expertise, we construct the UltraLink dataset for multilingual SFT. For each language, UltraLink consists of four subsets, encompassing chat data with language-specific content, chat data with language-agnostic content, math data, and code data.

To mitigate the challenge of data scarcity, some researchers suggest translating English SFT data into multiple languages. Lai et al. (2023) utilize ChatGPT[1] to translate the two essential components, instructions and responses, from Alpaca-style (Taori et al., 2023) English data to other languages. Chen et al. (2023) propose to translate both the Alpaca and the ShareGPT[2] data. While directly translating English SFT data can effectively

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

support multilingual SFT, there are still two major drawbacks associated with this approach:

- *Low cultural diversity and imprecise translations caused by cultural differences*: translation of English data may not adequately encompass topics specific to non-English regions (e.g., subjects related to Russian culinary culture), leading to a deficiency in language-specific knowledge for LLMs. Moreover, for certain instructions (e.g., `what are the most important holidays of the year?`), the answers vary in different cultural backgrounds, so directly translating all English conversations may result in numerous distorted translations.

- *Linearly increased data volume*: the total volume of translated SFT data linearly increases with the number of languages. However, the translations across different languages are semantically equivalent, making the model repeatedly learn the same content.

We believe that a good multilingual LLM should not only possess language-specific knowledge but also be equipped with language-agnostic skills. Figure 2 gives an example of the two types of instructions. We thus propose a new approach to better construct multilingual SFT data, applicable to any language. Compared to conversation translation (Lai et al., 2023; Chen et al., 2023), our advantages can be illustrated as follows:

- *Higher cultural diversity and less distorted translations*: for language-specific data, we propose a knowledge-grounded data augmentation method. Concretely, Wikipedia is employed[3] as a knowledge base for each language to provide more language-specific contexts. For language-agnostic chat data (e.g., the second example in Figure 2), we propose a two-stage translation mechanism. Given high-quality English SFT data, we first filter out the conversations that are specific to certain regions. Then we translate the remaining language-agnostic data.

- *Pruned data volume*: for language-agnostic skills like math reasoning and code generation, through our experiments, we find that it is unnecessary for the model to repeatedly learn identical problems, thanks to the strong cross-lingual transfer capabilities of modern LLMs. We can thus significantly prune the amount of math and code SFT data for non-English languages without compromising the model performance.

[3] https://www.wikipedia.org

1. Language-Specific Instructions
`What are some common tea traditions or etiquette observed in England?`
2. Language-Agnostic Instructions
`How do you approach learning a new skill or acquiring knowledge, and what strategies have you found to be effective in your learning process?`

Figure 2: Examples of instructions with language-specific and language-agnostic content.

We apply the aforementioned approach to four non-English languages, including Chinese, Russian, French, and Spanish. Note that our method can also be easily extended to other languages. Finally, we train an SFT LLM on the proposed UltraLink dataset, which outperforms several representative open-source multilingual LLMs, demonstrating the effectiveness of our dataset.

2 Data Curation

Automatically generating SFT data is now an important research topic for LLMs (Taori et al., 2023; Wang et al., 2023; Ding et al., 2023). For multilingual SFT, it is crucial to consider the influence of cultural diversity on language-specific data, while also integrating language-agnostic universal data that is related to the general ability of LLMs (i.e., math reasoning). In this work, we propose a data construction framework consisting of two pipelines, as shown in Figure 3.

2.1 Language-Specific Data Curation

The cultures around the world are vibrant and diverse, reflecting the lifestyles and perspectives of people from various countries and regions. To better cater to diverse users, the cultural diversity of multilingual LLMs should be improved. In this aspect, we propose a knowledge-grounded data augmentation method, leveraging language-specific knowledge bases to provide intricate and varied cultural backgrounds. Our method mainly contains two steps: (1) preparing and sampling knowledge from knowledge bases as cultural backgrounds, and

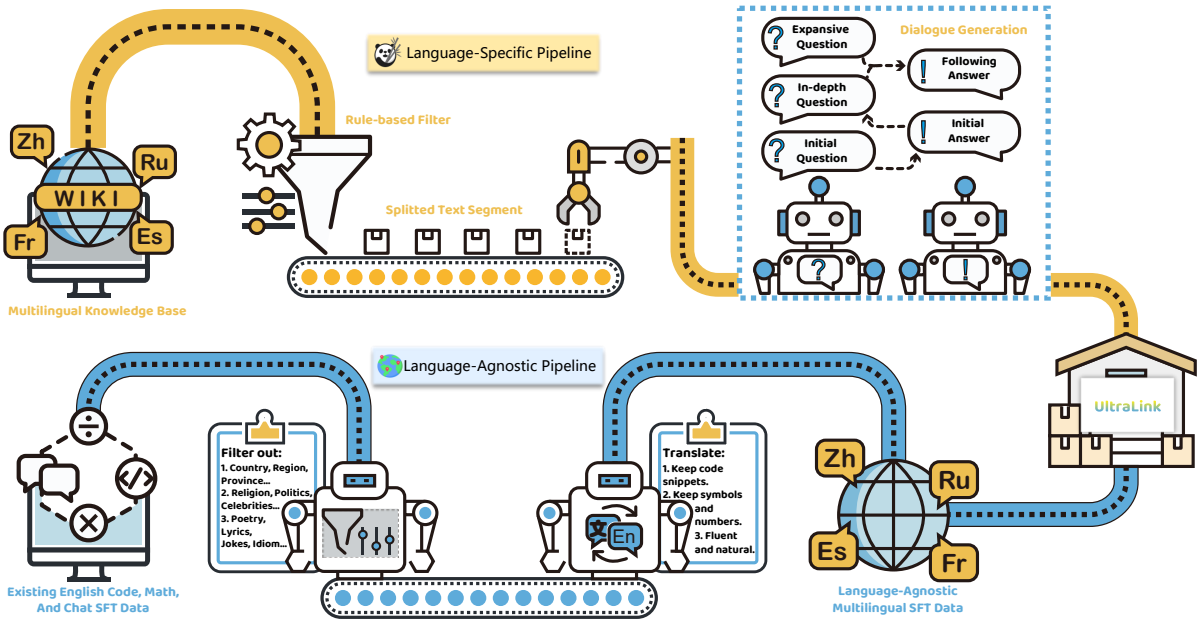

Figure 3: The proposed data augmentation method consists of two pipelines. The upper pipeline illustrates the generation of language-specific chat data. Dialogues are generated by LLMs, conditioning on language-specific knowledge extracted from Wikipedia. The language-agnostic pipeline aims to leverage existing high-quality English SFT data, using a two-stage translation mechanism to mitigate translation errors stemming from cultural differences.

(2) steering LLMs to generate informative conversations given the provided cultural backgrounds.

2.1.1 Knowledge Preparation

For each language, we utilize Wikipedia dumps[4] as the knowledge base, encompassing a diverse array of topics closely related to the respective culture. We first use an open-source extraction toolkit[5] to preprocess the raw dumps and get text descriptions for each entry. Then we use the language identification model provided by `fastText` (Joulin et al., 2017) to remove contents that are not in the expected language. For Chinese, we also use `OpenCC`[6] to convert traditional Chinese texts into simplified Chinese. Finally, we filter out documents that are shorter than 1K tokens or longer than 10K tokens. The number of tokens is calculated by `tiktoken`[7].

Given that most LLMs have a limited context length, we divide the whole text into segments whose lengths are between 1K and 2K. We do not split whole sentences when performing text segmentation. The preprocessed texts are used as contexts for the following dialogue generation procedure.

[4] https://dumps.wikimedia.org
[5] https://github.com/attardi/wikiextractor
[6] https://github.com/BYVoid/OpenCC
[7] https://github.com/openai/tiktoken

2.1.2 Dialogue Generation

To automatically generate multi-turn dialogues, we designed a question generator and an answer generator, which are both based on GPT-3.5. When generating the dialogue, both the question and answer generators are conditioned on a provided text segment as the cultural background. The used prompts can be divided into four parts: system prompt, principles, cultural background, and dialogue history. The prompt structure is shown in Figure 4. The system prompt is used to describe the task (i.e., `generating the initial question`). The principles provide some detailed suggestions for the LLM, which are found important for improving the quality of the generated data. The cultural background is the preprocessed text segment that contains language-specific knowledge. The dialogue history provides the historical questions and answers, which is set to an empty string when generating the initial question.

{system prompt} **{principles}**
<document> **{cultural background}** <\document>
{dialogue history}

Figure 4: Structure of the prompts used for dialogue generation. The provided cultural background is enclosed within a pair of separators.

Generating the Initial Dialogue The principles used to generate the first question are shown in Figure 5. We ask the involved LLM (i.e., GPT-3.5) to understand the provided cultural background and then propose a related question that can be answered according to the cultural background. For the generation of answers, we provide only a concise description of the principles in Figure 6 due to space limitations. For each language, the principles are translated by humans into the target language. We only show the English version of the prompt to better understand the method.

```
1. Pose "why" and "how" questions: given the provided document, ask why something happens or how it occurs. The questions should guide respondents to engage in more in-depth analysis and explanation, rather than simply stating facts.
2. Compare and contrast: if the text mentions a phenomenon or viewpoint, you can try comparing it with other similar situations and then pose questions to explore the similarities and differences between them, as well as potential impacts.
3. Predict future developments: if the text refers to a trend or direction of development, you can pose questions to discuss possible changes in the future or express opinions and predictions about a particular trend.
4. Stimulate reflection and discussion: Pose open-ended questions to encourage respondents to delve into deeper reflection and discussion.
```

Figure 5: Principles for generating the initial question.

```
1. Understand the content.
2. Logically reason about details.
3. Compare relevant situations.
4. Discuss future trends.
5. Engage in deeper discussion.
```

Figure 6: A brief description of the principles for generating the initial answer.

Generating Subsequent Dialogues After generating the initial question and answer, we iteratively produce subsequent dialogues. To improve the diversity of constructed dialogues, we propose two types of subsequent questions. At each turn, we randomly decide whether to present an *in-depth question* for a more detailed exploration of the same topic or to generate an *expansive question* to delve into other subjects. The principles used to ask an in-depth question are shown in Figure 7, while the principles used to ask an expansive question are shown in Figure 8. Note that when generating subsequent dialogues, the cultural background is also provided to the model. We will attach all the full prompts in supplementary materials.

```
1. Understand the context.
2. Uncover implicit information.
3. Challenge existing viewpoints.
4. Extend the topic.
5. Pose open-ended questions.
6. Delve into more complex logic.
```

Figure 7: A brief description of the principles to ask an in-depth following question.

```
1. Abstract the theme.
2. Turn into overarching topics.
3. Considering temporal and spatial span.
4. Connect to related fields.
5. Take a global perspective.
```

Figure 8: A brief description of the principles to ask an expansive following question.

Using the aforementioned approach, we automatically construct language-specific multi-turn conversations in four languages. The details of constructed data will be illustrated in Section 3, including the average length and some other statistics. Note that the proposed knowledge-grounded data augmentation approach can also be applied to any other language.

2.2 Language-Agnostic Data Crution

In addition to language-specific abilities, the general abilities that are language-agnostic are also essential for LLMs. As numerous high-quality English SFT datasets already encompass a broad spectrum of general abilities, we suggest employing a two-stage translation mechanism to maximize the utility of existing English resources. Our goal is to reduce translation errors caused by cultural differences since some questions can not be directly translated into other languages (e.g., `write an English poem where each sentence starts with the letter "A"`). In the first stage, we introduce a multi-criteria mechanism to filter out English-specific conversations that are difficult to translate accurately into other languages. Then we

use GPT-3.5 to translate the remaining language-agnostic data. In this study, we consider three key components of general abilities for LLMs: chat, math reasoning, and code generation. For chat, we use ShareGPT as the English chat data, which consists of multi-turn dialogues between human users and ChatGPT. For math reasoning, we use MetaMath (Yu et al., 2023) as the English math data. For code generation, we use the Magicoder dataset (Wei et al., 2023b) as the English code data.

2.2.1 Multi-Criteria Filter

The criteria employed to filter out English-specific conversations are outlined in Figure 9. Our goal is to retain only conversations whose topics can be discussed in any cultural background. GPT-3.5 is utilized to ascertain whether a conversation contains information relevant to the specified features. For instance, the conversations that include English jokes will be removed before translation.

```
1. Full name of *human*.
2. Country, region, state, province, city, address.
3. Conventions, politics, history, and religion.
4. Poetry, rhymes, myths, tales, jokes, and slang.
5. Food, cloth, furniture, construction.
6. Organization, company, product, brand.
```

Figure 9: Criteria used to identify English-specific conversation. We only provide a brief version with a detailed explanation due to space limitations.

2.2.2 Translator

After the filtering process, the remaining conversations undergo the translation procedure, wherein they are translated into four languages using GPT-3.5-turbo to maintain fluency and accuracy. We also provide some translation principles to help GPT-3.5 better perform the translation, which is shown in Figure 10.

2.3 Data Pruning

English math and code datasets are frequently extensive, exemplified by MetaMath (Yu et al., 2023) with 395K training examples and Magicoder (Wei et al., 2023b) comprising 186K training examples. Assuming the English data consists of N training examples, the overall multilingual dataset would encompass $k \times N$ examples if we translate all the English training examples into other languages,

```
1. Ensure the completeness and consistency of content during the translation process, without adding or deleting any information.
2. Ensure that the translated text is fluent and natural, using the most common expressions in the target language whenever possible. Use officially prescribed translations for professional terms and adhere to the target-language expression conventions.
3. If certain terms are not in natural language but are mathematical symbols, programming languages, or LaTex language, please directly copy the original text.
4. If there are no equivalent translation terms for certain vocabulary, please directly copy the original text.
5. For citations and references, please directly copy the original text.
```

Figure 10: Translation principles.

where k is the number of languages. The linear increase in data volume will result in higher training costs during SFT. As math and code problems are not closely tied to the cultural backgrounds of different countries, LLMs may have the capability to transfer English math and code abilities into other languages with only limited training examples. In other words, it may not be necessary for LLMs to learn all translated math and code problems. To verify the assumption mentioned above, we conduct experiments on Chinese math and code tasks. For comparison, we fine-tune Llama-2-7b (Touvron et al., 2023b) in the following two different ways:

- *From En SFT Model*: we first use English math or code data to fine-tune the base model, and then use different amounts of Chinese data to further tune the model.
- *From Base Model*: we directly use Chinese math or code data to fine-tune the base model.

Figure 11 and 12 show the performances of the two types of models. Surprisingly, the involved LLM exhibits strong cross-lingual transfer capabilities. For instance, utilizing only 2K Chinese mathematical training examples can yield a score of 45.6 when fine-tuning from the English SFT model. In contrast, directly fine-tuning the base model with an equivalent amount of Chinese data results in a significantly lower score of 22.0, highlighting the superior performance achieved through transfer from the English SFT model. In the Chinese

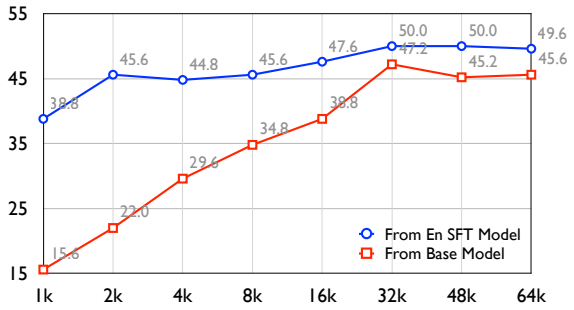

Figure 11: Performance on MGSM-Zh with different numbers of Chinese mathematical training examples.

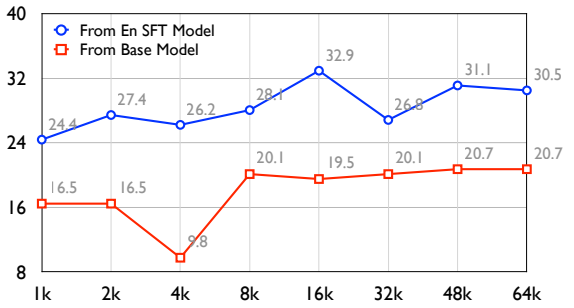

Figure 12: Performance on HumanEval-Zh with different numbers of Chinese code training examples.

code generation task, we observe a similar trend, wherein transfer learning from the English SFT model substantially enhances the performance of the model.

Moreover, we find that using more Chinese SFT data does not consistently lead to improved performance. For the math task, using 32K Chinese training examples achieves the best result. For the code task, the peak performance is attained with 16K Chinese code generation examples. Hence, we incorporate only 32K mathematical training examples and 16K code training examples for each non-English language in the UltraLink dataset.

Lang.	Lang.Spec.	Lang.Agno.		
	Chat	Chat	Math	Code
En	10K	67K	395K	186K
Zh	36K	11K	32K	16K
Ru	37K	11K	32K	16K
Fr	30K	11K	32K	16K
Es	34K	11K	32K	16K
UltraLink	147K	112K	523K	250K
w/o En	137K	45K	128K	64K

Table 1: Scales of different components in UltraLink, which are measured by the number of dialogues.

3 Dataset Statistics

3.1 Data Distribution

Table 1 presents the scale of each component in UltraLink, encompassing five languages. Each language contributes four types of SFT data: chat data with language-specific knowledge, chat data with language-agnostic knowledge, math data, and code data. The quantities of language-agnostic segments are approximately equal for the four non-English languages.

3.2 Comparison with Existing Datasets

Before us, there are some existing multilingual SFT datasets, where we select four representative datasets for comparison, including the Okapi dataset (Lai et al., 2023), the Guanaco dataset (Attardi, 2023), Multialpaca (Wei et al., 2023a), and the Phoenix SFT data (Chen et al., 2023). We conduct a comparison based on the number of dialogues, the number of conversation turns, and the average lengths across the respective datasets. As shown in Table 2, we find that UltraLink contains fewer dialogues than the Guanaco dataset, but the latter only contains single-turn conversations. Only the Phoenix SFT data and UltraLink include multi-turn conversations.

We use the number of tokens estimated by `tiktoken` as the length for each question and answer. The question token length does not include the document. On average, UltraLink exhibits the longest average length per turn (i.e., 378.21 tokens), considering both questions and their corresponding answers. Compared to UltraLink, the Phoenix SFT data has longer questions (165.27 vs. 87.86), but its answers are shorter (200.07 vs. 290.35).

For each language, we also estimate the average lengths of questions and answers, and the results are shown in Figure 13. Across all languages, the answer is significantly longer than the question.

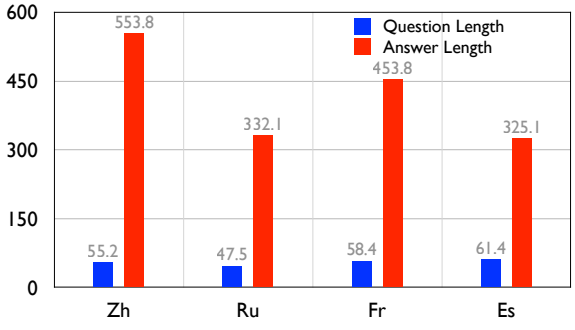

Figure 13: Number of tokens for each language.

Dataset	Dialogues	Turns	Average Length		
			Question	Answer	Turn
Okapi Dataset (Lai et al., 2023)	207K	207K	28.64	95.72	124.36
Guanaco Dataset (Attardi, 2023)	**1173K**	1173K	77.58	83.31	160.89
Multialpaca (Wei et al., 2023a)	132K	132K	39.86	83.71	123.57
Phoenix SFT data (Chen et al., 2023)	464K	893K	**165.27**	200.07	365.34
UltraLink (**Ours**)	1032K	**1623K**	87.86	**290.35**	**378.21**

Table 2: Comparison between UltraLink and existing open-source multilingual SFT datasets.

4 Experiment

4.1 Setup

Baselines For thorough comparison, we select several representative multilingual baselines in our experiments, including Bloomz-7b1-mt (BigScience, 2023), Phoenix-inst-chat-7b (Chen et al., 2023), PolyLM-Multialpaca-13b (Wei et al., 2023a), PolyLM-Chat-13b (Wei et al., 2023a), Chimera-inst-chat-13b (Chen et al., 2023), Okapi-7b (Lai et al., 2023), Guanaco-7b (Attardi, 2023), and Guanaco-13b (Attardi, 2023). Okapi-7b is fine-tuned by ourselves based on Llama-2-7b using the Okapi dataset, while other baselines are downloaded from Huggingface[8].

Training details Based on Llama-2-13b (Touvron et al., 2023a), UltraLink-LM is fine-tuned with the constructed UltraLink dataset for 3 epochs. We use the cosine learning rate schedule and the peak learning rate is set to 2e-5. The warm-up ratio is set to 0.04. Each mini-batch contains 128 training examples in total. The maximum sequence length is 4096. We train the model using 32 A100 GPUs for about 140 hours.

Evaluation We examine the model performance on three tasks, including chat, math reasoning, and code generation. For chat, we use OMGEval (Liu et al., 2023) for evaluation, which is a multilingual version of the widely-used English benchmark AlpacaEval (Li et al., 2023). OMGEval is not a mere translated version of AlpacaEval. Instead, it localizes the English questions according to the cultural backgrounds of each language. We employ MGSM (Shi et al., 2023) to evaluate math reasoning abilities, which is also a multilingual benchmark. Since there are no existing multilingual test sets for code generation, we use GPT-3.5 with carefully designed prompts to translate HumanEval (Chen et al., 2021) into other languages, which serves as the multilingual benchmark to evaluate the code abilities of LLMs. We use the `UltraEval` toolkit[9] for model inference and evaluation, which supports a wide range of open-source models.

4.2 Results

Table 3 shows the results of the involved multilingual SFT LLMs on different tasks. In terms of general chat abilities, our model achieves the best average results. While Guanaco-13b slightly outperforms us in English (29.0 vs. 28.8), its performance is notably lower than ours in non-English languages. Given that Guanaco-13b shares the same backbone (i.e., Llama-2-13b) with our model, the results imply the superiority of the proposed UltraLink dataset.

For the code generation Task, previous multilingual SFT datasets did not take into account the multilingual code abilities, which we think is very important in many real-world scenarios. Our model achieves a score of 60.4 in the English HumanEval benchmark, surpassing even CodeLlama-34b-Python (Rozière et al., 2024). For comparison, training the model solely on the English Magicoder (Wei et al., 2023b) dataset results in a HumanEval score of 53.0. The improvement of UltraLink-LM over the model trained on the English Magicoder dataset (i.e., 60.4 vs. 53.0) suggests that the constructed code SFT data in other languages can also enhance English code abilities. This confirms our assumption that modern LLMs possess strong transfer abilities for language-agnostic skills.

In the math reasoning task, our model consistently outperforms all other baselines across all five languages. The performance of UltraLink-LM in both math and code tasks underscores the effectiveness of our method in enabling multilingual LLMs to acquire general abilities.

[8] https://huggingface.co

[9] https://github.com/OpenBMB/UltraEval

Model	Backbone	SFT Data	OMGEval (Chat)					
			En	Zh	Es	Ru	Fr	Avg.
Bloomz-7b1-mt	Bloomz-7b1	xP3mt	0.0	0.9	0.1	0.5	0.3	0.4
Phoenix-inst-chat-7b	Bloomz-7b1	Phoenix SFT data	6.9	13.3	7.4	2.9	8.1	7.7
PolyLM-Multialpaca-13b	PolyLM-13b	Multialpaca	3.4	5.0	2.1	5.1	2.2	3.6
PolyLM-Chat-13b	PolyLM-13b	Closed-source	7.7	14.0	6.1	5.5	4.8	7.6
Chimera-inst-chat-13b	Llama-13b	Phoenix SFT data	15.5	9.7	11.8	13.7	13.8	12.9
Okapi-7b	Llama-2-7b	Okapi Dataset	8.8	6.2	5.0	12.1	8.7	8.2
Guanaco-7b	Llama-2-7b	Guanaco Dataset	4.6	3.8	0.4	1.8	1.2	2.4
Guanaco-13b	Llama-2-13b	Guanaco Dataset	**29.0**	8.6	16.9	15.4	17.3	17.5
UltraLink-LM	Llama-2-13b	UltraLink	28.8	**21.9**	**23.5**	**37.6**	**29.0**	**28.2**
Model	**Backbone**	**SFT Data**	**Multilingual HumanEval (Code)**					
			En	Zh	Es	Ru	Fr	Avg.
Bloomz-7b1-mt	Bloomz-7b1	xP3mt	8.5	7.3	6.1	8.5	6.1	7.3
Phoenix-inst-chat-7b	Bloomz-7b1	Phoenix SFT data	11.0	10.4	8.5	1.2	13.4	12.2
PolyLM-Multialpaca-13b	PolyLM-13b	Multialpaca	8.5	7.3	6.1	6.1	6.1	6.8
PolyLM-Chat-13b	PolyLM-13b	Closed-source	10.4	7.9	6.1	7.3	8.5	8.1
Chimera-inst-chat-13b	Llama-13b	Phoenix SFT data	14.6	13.4	14.6	12.8	14.0	13.9
Okapi-7b	Llama-2-7b	Okapi Dataset	12.2	11.0	8.5	8.5	8.5	9.8
Guanaco-7b	Llama-2-7b	Guanaco Dataset	9.2	6.7	11.0	9.8	12.8	9.9
Guanaco-13b	Llama-2-13b	Guanaco Dataset	18.3	15.9	9.8	8.5	14.6	12.2
UltraLink-LM	Llama-2-13b	UltraLink	**60.4**	**43.9**	**40.9**	**49.4**	**39.6**	**46.8**
Model	**Backbone**	**SFT Data**	**MGSM (Math)**					
			En	Zh	Es	Ru	Fr	Avg.
Bloomz-7b1-mt	Bloomz-7b1	xP3mt	2.8	1.6	2.0	0.4	2.8	1.7
Phoenix-inst-chat-7b	Bloomz-7b1	Phoenix SFT data	3.2	3.2	2.8	3.2	3.2	3.1
PolyLM-Multialpaca-13b	PolyLM-13b	Multialpaca	1.2	2.8	1.6	2.8	2.4	2.4
PolyLM-Chat-13b	PolyLM-13b	Closed-source	10.8	6.4	4.8	4.4	5.6	5.3
Chimera-inst-chat-13b	Llama-13b	Phoenix SFT data	14.0	11.6	10.0	12.0	12.8	11.6
Okapi-7b	Llama-2-7b	Okapi Dataset	4.0	2.4	3.6	4.4	4.8	3.8
Guanaco-7b	Llama-2-7b	Guanaco Dataset	4.0	1.6	3.2	2.8	4.4	3.0
Guanaco-13b	Llama-2-13b	Guanaco Dataset	13.6	10.8	11.2	6.4	5.2	8.4
UltraLink-LM	Llama-2-13b	UltraLink	**70.4**	**56.0**	**70.4**	**64.8**	**63.6**	**63.7**

Table 3: Performance of the involved multilingual SFT LLMs on different tasks.

5 Related Works

Supervised Fine-tuning SFT is now a crucial part of constructing a powerful LLM. SODA (Kim et al., 2023) constructs high-quality social dialogues by contextualizing social commonsense knowledge from a knowledge graph. Using the technique of self-instruct (Wang et al., 2023), Alpaca (Taori et al., 2023) is one of the pioneers to leverage ChatGPT to collect SFT data. UltraChat (Ding et al., 2023) utilizes ChatGPT to generate topics in a tree-style structure for the construction of large-scale dialogues. With these efforts, English SFT resources are becoming increasingly rich and effective.

Multilingual SFT Datasets To enhance the global utility of LLMs, numerous multilingual SFT datasets have been created. Lai et al. (2023) employ ChatGPT to translate Alpaca into various languages. Chen et al. (2023) combine ShareGPT with Alpaca and then translate the two datasets. Attardi (2023) and Wei et al. (2023a) extend tasks from Alpaca by introducing filters and rewrites of seed tasks in different languages, generating datasets through multiple iterations. This work proposes the utilization of a multilingual knowledge base to enhance the cultural diversity of multilingual Supervised Fine-Tuning data, as well as to improve the language-agnostic general abilities of LLMs through cross-lingual transfer learning.

6 Conclusion

In this work, we propose a knowledge-grounded data augmentation method and a two-stage translation mechanism to construct language-specific and language-agnostic multilingual SFT data, respectively. Experiments demonstrate that the proposed dataset is effective for multilingual LLMs.

7 Ethical Impact

We present a framework for generating SFT data across diverse languages and use the proposed dataset to learn an LLM. Our LLM may inevitably encounter common challenges, including issues such as hallucination and toxicity. We highly recommend users utilize our work exclusively for research purposes, to enhance the efficacy of LLMs across various languages.

8 Limitations

In the paper, our proposed data construction framework is only applied to four language types. Nevertheless, the framework can be easily extended to other languages. We leave it to the future work to include more languages. Moreover, due to constraints imposed by the base model, the multilingual capability still faces several limitations. Notably, the model exhibits significantly better performance in English across many tasks. There is a pressing need to continue constructing high-quality pre-training multilingual datasets, to unlock the full potential of multilingual abilities in LLMs.

References

Yuvanesh Anand, Zach Nussbaum, Adam Treat, Aaron Miller, Richard Guo, Ben Schmidt, GPT4All Community, Brandon Duderstadt, and Andriy Mulyar. 2023. Gpt4all: An ecosystem of open source compressed language models.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Giusepppe Attardi. 2023. Guanaco. https://guanaco-model.github.io/.

BigScience. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. Phoenix: Democratizing chatgpt across languages. *ArXiv*, abs/2304.10453.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*.

Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2023. SODA: Million-scale dialogue distillation with social commonsense contextualization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

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

Yang Liu, Lin Zhu, Jingsi Yu, Meng Xu, Yujie Wang, Hongxiang Chang, Jiaxin Yuan, Cunliang Kong, Jiyuan An, Tianlin Yang, Shuo Wang, Zhenghao Liu, Yun Chen, Erhong Yang, Yang Liu, and Maosong Sun. 2023. Omgeval: An open multilingual generative evaluation benchmark for foundation models. `https://github.com/blcuicall/OMGEval`.

OpenAI. 2023. Gpt-4 technical report.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code.

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. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Xiangpeng Wei, Hao-Ran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, Tianxiang Hu, Shangjie Li, Binyuan Hui, Yu Bowen, Dayiheng Liu, Baosong Yang, Fei Huang, and Jun Xie. 2023a. Polylm: An open source polyglot large language model. *ArXiv*, abs/2307.06018.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023b. Magicoder: Source code is all you need.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models.