Task Analyzing . After rephrasing the task input, we need to complete the task in task goal. This step is comparable to the task description used in conventional prompting methods. In practice, we can get the task information from the literature or seek assistance from ChatGPT to generate effective prompts for solving the task (Jiao et al., 2023).

CoT Task Solving . We then ask the model to follow the instructions and complete the task step by step. Since LLMs exhibit a strong ability to maintain a chain-of-thought (Wei et al., 2022c), we carefully design instructions to guide the model, with the hope that it will respond to our instructions in a step-by-step manner and utilize the intermediate outputs to aid in solving the task.

Output Formatting . Finally, we should regularize the output format of the model to obtain the exact answer. LLMs are utilized in a zero- or few-shot manner, and they tend to generate texts that may not conform to the format of the target answer. Fortunately, LLMs possess a strong ability to follow instructions, and we can define the output format in terms of output type and output constraint. The output type can be a number, index, or text, while the output constraint is optional and determined based on the task requirements. Output constraint may include length limitations, language specifications, and other relevant factors.

2.2 XLT for Few-shot Learning

The above construction of XLT can be directly fed to LLMs to yield outputs, which is performed in the zero-shot learning setting. In addition, we also explore incorporating demonstrations into XLT to enable few-shot learning. Different from previous work that just appends model outputs to the corresponding request (Shi et al., 2023) or utilizes a verbalizer to format the output, our method constructs the demonstrations with better formatted model outputs from a step-by-step processing-based XLT. As illustrated in Figure 4, we first sample a few examples from the development set and incorporate the requested parts into XLT. The zero-shot learning is performed over LLM to collect responses that are further aligned with those of the samples. Only response-aligned requests are assembled with the corresponding model responses to form final demonstrations for few-shot learning. In this way, the demonstrations are constructed with rich logical knowledge via XLT, which will cater to the XLT-based generation of new requests. In practice, we can also correct or design the demonstrations for better alignment with the instruction logic.

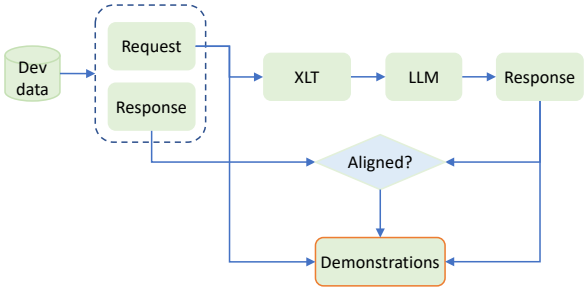

Figure 4: Construction process for few-shot learning.

3 Experiments

To comprehensively verify the effectiveness of our method on language-independent generality, we evaluate our XLT template on different LLMs covering various natural language processing tasks in multiple languages.

3.1 Experimental Setups

3.1.1 Tasks and Benchmarks

We conduct evaluations on seven typical benchmarks related to reasoning, understanding, and generation tasks that can represent different capabilities of LLMs, encompassing both high-resource and low-resource languages. These benchmarks cover 27 different languages, including English (en), German (de), Russian (ru), French (fr), Chinese Simplified (zh), Spanish (es), Japanese (ja), Italian (it), Vietnamese (vi), Turkish (tr), Indonesian (id), Swahili (sw), Arabic (ar), Korean (ko), Greek (el), Thai (th), Bulgarian (bg), Hindi (hi), Estonian (et), Bengali (bn), Tamil (ta), Galician (gl), Urdu (ur), Telugu (te), Javanese (jv), Haitian Creole (ht), and Southern Quechua (qu). In terms of the language distribution statistics in the Common Crawl Monthly Archives[3] and the language performance of LLMs (Shi et al., 2023; Ahuja et al., 2023), we have arranged them in the order of language frequency from high-resource to low-resource. In particular, the frequency of some underrepresented languages is even less than 0.1% (*e.g.,* bn, ta, gl, ur, te, jv, ht, qu).

- **Reasoning tasks**

– **Arithmetic Reasoning**. The MGSM (Shi et al., 2023) benchmark contains grade school mathe-

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

matical problems and asks the model to calculate the correct answer. It covers 11 languages, and we utilize the accuracy score for evaluation.

– **Commonsense Reasoning**. The XCOPA (Ponti et al., 2020) benchmark contains one premise and two choices. It asks the model to choose which one is the result or cause of the premise. It covers 11 languages from 11 diverse families, and we utilize the accuracy score for evaluation.

• **Understanding tasks**

– **Natural Language Inference**. The XNLI (Conneau et al., 2018) benchmark contains one premise and one hypothesis and requires the model to determine whether the hypothesis is entailed, contradicted, or neutral conditioned on the premise. It covers 15 languages, and we utilize the accuracy score for evaluation.

– **Paraphrase Identification**. The PAWS-X (Yang et al., 2019) benchmark contains two sentences and requires the model to judge whether they paraphrase each other or not. It covers 7 languages, and we utilize the accuracy score for evaluation.

• **Generation tasks**

– **Question Answering**. The MKQA (Longpre et al., 2021) benchmark contains an open-domain question and asks the model to predict a short answer. Since it has unanswerable questions or long questions that do not have precise answers, we remove these questions during evaluation. It covers 25 languages, and we choose a subset of 10 languages, including de, en, es, fr, ja, ru, th, tr, vi, and zh. We utilize the token overlap F1 score for evaluation.

– **Summarization**. The XL-Sum* (Hasan et al., 2021) (250 test samples randomly sampled from XL-Sum per language) benchmark contains a long news article and wants the model to summarize it into a short text. It covers 44 languages, and we choose a subset of 6 languages, including en, es, fr, tr, vi, and zh. We utilize the ROUGE-1 score (Lin, 2004) for evaluation.

– **Machine Translation**. The FLORES* (Costa-jussà et al., 2022) (200 test samples randomly sampled from FLORES-200 per language) benchmark contains parallel text from Wikimedia projects for 204 languages, yielding over 40,000 translation directions. We choose a subset of 12 directions, including high resource to high resource translation (*i.e.,* zh↔ ru and de↔vi), high resource to low resource translation (*i.e.,* zh↔ th and zh↔jv), and low resource to low resource translation (*i.e.,* th↔gl and jv↔th). We utilize the SacreBLEU score (Papineni et al., 2002; Post, 2018) for evaluation.

Among these benchmarks, MGSM, XCOPA, XNLI, PAWS-X, and MKQA are parallel, *i.e.,* the instances are semantics-equivalent across each language. For all benchmarks, we report the results on the test sets using all instances (Table 5), except for XL-Sum and FLORES-200, where we only sample 250 and 200 examples respectively to show the trend of generation performance. In the few-shot setting, we randomly choose examples from the development set if they have, otherwise, we translate the English training set into corresponding languages to construct several examples.

3.1.2 Baselines

Basic Prompt are the vanilla in our experiments that were proposed and suggested in previous work. After determining the prompt, we format each monolingual instance using the English basic prompt. This setting is similar to the monolingual prompting in MEGA (Ahuja et al., 2023). The basic prompts used for the evaluation of each benchmark are listed in Table 5. Note that, we dismiss the baseline using native-language, since MEGA (Ahuja et al., 2023) reveals monolingual prompting is superior to cross-lingual prompting.

Chain-of-Thought (CoT) prompting invokes LLMs to generate a series of intermediate results to solve reasoning tasks (Wei et al., 2022c), which is still effective under multilingual scenarios (Shi et al., 2023). In experiments, we append the instruction *"Let's think step-by-step and tell me the answer in the end"* after the input to prompt LLMs.

Translate-English leverages the robust capabilities of LLMs in English to tackle multilingual tasks, as suggested by both Shi et al. (2023) and Ahuja et al. (2023). This approach translates instances from other languages into English beforehand. In practice, we utilize the Google Translate API to translate examples into English and apply the basic prompt to format them. Note that, we do not apply this method to generation tasks since they require the output in respective language rather English.

XLT utilizes the proposed template consisting of multiple instructions introduced in Section 2. The

instantiated XLT templates for each benchmark are listed in Table 6.

In few-shot learning scenarios, for basic prompt, we use the same template as an additional input to the model. For XLT, we provide the exemplars with XLT template inputs and anticipate desirable step-by-step outputs as outlined in Figure 4. In the subsequent evaluation, we apply the 5-shot setting, except for the XL-Sum* experiments, which use the 3-shot setting due to input length constraints.

3.1.3 LLMs

We mainly evaluate two LLMs from the GPT-3.5 series models:

- `text-davinci-003`[4] is trained using instruction tuning and reinforcement learning from human feedback (Ouyang et al., 2022). It can perform a wide range of natural language tasks with satisfactory results.
- `gpt-3.5-turbo`[4] is optimized for chat based on `text-davinci-003` and suitable for traditional NLP tasks. It is the most capable GPT-3.5 model.

To verify the compatibility of our XLT template, we further incorporate LLaMA-2-Chat (Touvron et al., 2023) (`Llama-2-70b-chat-hf`) as our base models. It is an open-source model that has been trained through supervised fine-tuning and reinforcement learning from human feedback on the base LLaMA 2 model. In addition, we also refer to the existing results from other LLMs, such as `code-davinci-002`[4], when the evaluation is comparable. During inference, we employ greedy search (*i.e.,* temperature=0) to generate the LLM responses. We find LLMs have excellent instruction-following abilities to respond to our instructions in the given format. Therefore, we just extract the part after "Answer format:" as labels.

3.2 Experimental Results

Multilingual Capability. We comprehensively evaluate XLT's performance over seven tasks. The average score of `text-davinci-003` is summarized in Figure 1(a) and Table 1, and more details are listed in Appendix A. As for the CoT prompting, it can enhance reasoning tasks while becomes less effective on understanding and generation tasks. In terms of the Translate-En prompting, it can boost the performance in the zero-shot settings while may not work well in the few-shot settings. Overall, compared to the three baseline methods, XLT achieves significant improvements over two LLMs for all tasks on both zero-shot and few-shot settings regardless of the language difference, except for a slight drop on the PAWS-X benchmark in the zero-shot setting. It is noted that XLT achieves remarkable gains of nearly 20 points on average in the MGSM benchmark for the arithmetic reasoning task and around 10 points on average in the MKQA benchmark for the open-domain question answering task. The experiments demonstrates the effectiveness of XLT for empowering LLM with multilingual capability.

As for the compatibility test, we list the results of LLaMA-2-Chat on the MGSM benchmark in Table 7. It is notable that LLaMA 2 can also benefit from our cross-lingual-thought, which further demonstrates the generality of our XLT template. However, the gains of LLaMA-2-Chat is not as good as GPT-based models. Our analysis reveals this gap can primarily be attributed to LLaMA 2's poorer multi-step instruction-following ability.

Language Democratization. Furthermore, we try to assess the democratization degree of tasks between languages by defining a "democratization score", which calculates the average percentage of performance attained by different languages relative to the best performance among all languages. Given the evaluation scores of $s_1, s_2, \ldots, s_l$ corresponding to l language on a task, the democratization score is formulated as:

$$\frac{\sum_{i=1}^{l} s_i}{l} / \max\{s_i\}_{i=1}^{l}. \quad (1)$$

Table 2 presents the degree of democratization for tasks across languages under both zero-shot learning and few-shot learning, and we further summarize it in Figure 1(b) by averaging all scores per task regardless of the setting and model differences. We can observe that XLT leads to higher democratization scores in general, particularly for XCOPA, and MKQA. As for MGSM, XNLI, and PAWS-X, our XLT can improve performance in multiple languages, where the overall performance of the baseline is consistently lower but the gap between languages is smaller as shown in Tables 7, 9, and 10. In conclusion, our method can reduce the performance gap between languages and improve the language democratization of LLMs.

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