Enhancing Cross-lingual Prompting with Two-level Augmentation

Anonymous ACL submission

Abstract

Prompting approaches show promising results in few-shot scenarios. However, its strength for multilingual/cross-lingual problems has not been fully exploited. Zhao and Schütze (2021) made initial explorations in this direction by presenting that cross-lingual prompting outperforms cross-lingual finetuning. In this paper, we first conduct sensitivity analysis on the effect of each component in cross-lingual prompting and derive Universal Prompting across languages. Based on this, we propose a two-level augmentation framework to further improve the performance of prompt-based cross-lingual transfer. Notably, for XNLI, our method achieves 46.54% with only 16 English training examples per class, significantly better than 34.99% of finetuning.

1 Introduction

Although adapting Pre-trained Language Models (PLMs) (Devlin et al., 2019) to downstream NLP tasks via finetuning is the de facto mainstream paradigm under fully supervised settings (Wang et al., 2018), prompting (Liu et al., 2021; Lester et al., 2021; Radford et al., 2019; Brown et al., 2020) has demonstrated its superiority to finetuning in low-resource scenarios (Schick and Schütze, 2021a,b), where the annotated training data is scarce or even not available. Typically, prompting reformulates the classification task as a language modeling problem over manually-designed natural language prompts.

Despite the effectiveness of prompting on English tasks, its potential for cross-lingual and multilingual problems, which assume the availability of the training data in high-resource languages (e.g., English) only, is still under-explored. Zhao and Schütze (2021) is the pioneering work to apply prompting to cross-lingual NLP. However, their major efforts are spent on comparing different training strategies for cross-lingual prompting, and how the key ingredients of prompting, namely prompt-design and inference strategies, affect the cross-lingual transfer is not discussed.

To provide a practical guide for cross-lingual prompting, we conduct a sensitivity analysis upon Zhao and Schütze (2021) to explore the effects of each prompting component on the performance of cross-lingual transfer. Surprisingly, in contrast to the complicated designs in Zhao and Schütze (2021), we find that neither template translation nor verbalizer translation for inference is necessary, and the template-free prompting coupled with English-only inference, dubbed as “Universal Prompting” in this paper, generally performs well across different few-shot settings.

Based on such findings, we further propose a two-level augmentation framework to enhance the performance of cross-lingual prompting. Specifically, motivated by the fact that there is no explicit target-language guidance in Universal Prompting, we firstly propose to utilize multilingual verbalizers as an answer augmentation approach. Multilingual verbalizers introduce the label tokens in target languages, which provides additional supervision signals for prompting. By doing so, the model is enforced to learn the association between prompts and semantically equivalent label tokens in multiple languages. Besides, to alleviate the data scarcity issue in few-shot settings, we also develop in-batch data augmentation, which is based on mixup (Zhang et al., 2018; Sun et al., 2020) mechanism, to enhance the training without additional unlabeled data (Xie et al., 2020) or efforts on text manipulation (Wei and Zou, 2019).

In summary, our contributions are as follows:

• We develop a simple yet effective baseline called Universal Prompting for cross-lingual prompting.
• Based on Universal Prompting, we further propose a two-level augmentation framework to enhance the performance of prompt-based cross-lingual transfer.
Table 1: Prompt templates and verbalizers in English (EN) and Turkish (TR). A and B indicate two sentences of a sentence pair. For XNLI, A is the premise and B is the hypothesis. With the proposed Universal Prompting, we could treat source-language training and target-language inference in a unified fashion.

<table>
<thead>
<tr>
<th>EN (source)</th>
<th>Prompt Templates</th>
<th>TR (target)</th>
<th>Verbalizers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Prompting</td>
<td>A. B ? &lt;mask&gt;.</td>
<td></td>
<td>Entailment: Evet; Contradict: hiçbir; Neutral: belki</td>
</tr>
<tr>
<td>w/o Prompting Words</td>
<td>Universal Prompting</td>
<td></td>
<td>Entailment: yes; Contradict: no; Neutral: maybe</td>
</tr>
<tr>
<td>w/o Verbalizer Translation</td>
<td></td>
<td></td>
<td>Entailment: yes; Contradict: no; Neutral: maybe</td>
</tr>
<tr>
<td>w/o Template Translation</td>
<td></td>
<td></td>
<td>Entailment: yes; Contradict: no; Neutral: maybe</td>
</tr>
</tbody>
</table>

Table 2: The comparison results between Zhao and Schütze (2021) and its variants on XNLI. We calculate the average accuracy over 15 languages. The standard deviation over 5 runs is reported as the subscript.

<table>
<thead>
<tr>
<th>Shots</th>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Zhao and Schütze (2021)</td>
<td>38.811.61</td>
</tr>
<tr>
<td></td>
<td>w/o Prompting Words</td>
<td>39.151.73</td>
</tr>
<tr>
<td></td>
<td>w/o Verbalizer Translation</td>
<td>42.321.81</td>
</tr>
<tr>
<td></td>
<td>w/o Template Translation</td>
<td>39.872.94</td>
</tr>
<tr>
<td></td>
<td>Universal Prompting</td>
<td><strong>43.182.77</strong></td>
</tr>
<tr>
<td>32</td>
<td>Zhao and Schütze (2021)</td>
<td>41.421.66</td>
</tr>
<tr>
<td></td>
<td>w/o Prompting Words</td>
<td>41.721.89</td>
</tr>
<tr>
<td></td>
<td>w/o Verbalizer Translation</td>
<td>46.501.54</td>
</tr>
<tr>
<td></td>
<td>w/o Template Translation</td>
<td>43.660.96</td>
</tr>
<tr>
<td></td>
<td>Universal Prompting</td>
<td><strong>48.261.34</strong></td>
</tr>
<tr>
<td>64</td>
<td>Zhao and Schütze (2021)</td>
<td>46.420.65</td>
</tr>
<tr>
<td></td>
<td>w/o Prompting Words</td>
<td>46.730.61</td>
</tr>
<tr>
<td></td>
<td>w/o Verbalizer Translation</td>
<td>53.071.33</td>
</tr>
<tr>
<td></td>
<td>w/o Template Translation</td>
<td>47.601.09</td>
</tr>
<tr>
<td></td>
<td>Universal Prompting</td>
<td><strong>52.191.53</strong></td>
</tr>
</tbody>
</table>

2.1 Universal Prompting across Languages

Zhao and Schütze (2021) achieved prompt-based cross-lingual transfer by directly utilizing the translated prompting words and verbalizers for target-language inference. However, since the translated prompting words are not seen and the translated verbalizers are never modeled by the PLM during training on English, this may result in discrepancies between the source-language training and the target-language inference.

Starting from the above two aspects that result in such source-target discrepancies, we consider 3 possible variants with design choices different from Zhao and Schütze (2021) to alleviate the discrepancies to a certain degree. By combining these variations we end up with a Universal Prompting design, which can treat individual languages in a unified fashion. Table 1 summarizes our different design choices. 1

2.2 Results

Our major experimental setup follows Zhao and Schütze (2021). Please refer to Section 4 for more details. In Table 2, we show that by alleviating discrepancies either in the aspect of verbalizers or templates, we could further improve the performance of cross-lingual prompting. Our proposed Universal Prompting across languages alleviates the discrepancy of prompt templates and verbalizers simultaneously, yielding a much stronger baseline than Zhao and Schütze (2021). This indicates that a null prompt (IV et al., 2021), combined with the English verbalizer for target-language inference generally performs well in multilingual tasks. We refer to this design as Universal Prompting (UP) in the following parts of our paper.

3 Method

Mask token in prompting methods is directly used for inference. In this section, we formalize our two-level augmentation approach for this important element of prompting. Our method leverages answer-level multilingual verbalizers and representation-level mixup simultaneously.

3.1 Answer-level Multilingual Verbalizers

The derived UP only considers the English verbalizer for source language training, and the translated
verbalizers in target languages are not exploited. Intuitively, their rich semantics could serve as high-quality paraphrases (Jiang et al., 2021) of the English verbalizer and provide additional supervision for training multilingual models. Motivated by this, we define a multilingual verbalizer for the English training data, which can be regarded as answer-level augmentation for masked language modeling. Formally, given the pre-built prompt \( x \) filled with input sentences, the training objective is to maximize the likelihood of verbalized label tokens in multiple languages:

\[
\arg\max_{\theta} \frac{1}{|L|} \sum_{\ell \in L} \log P(\langle \text{mask} \rangle = V(\ell)(x; \theta))
\]

where \( \theta \) denotes parameters of the PLM, \( V(\ell) \) is the verbalizer in a certain language \( \ell \in L \), and it maps from the gold label to a specific word in language \( \ell \). In comparison, UP only takes \( L = \{\text{EN}\} \), which is a monolingual verbalizer.

### 3.2 Representation-level Mixup

Manifold mixup (Verma et al., 2019) performs the interpolation in the latent space to construct virtual labeled data as augmentation. Based on manifold mixup, several mixup strategies have been designed to boost the performance of NLP tasks (Chen et al., 2020; Sun et al., 2020; Zhang and Vaidya, 2021).

In this work, we propose to use mixup for cross-lingual prompting as a representation-level augmentation approach. To the best of our knowledge, this is the first endeavor to enhance prompting and multilingual learning with mixup. To formalize, let \( m_i = h(x_i) \) and \( m_j = h(x_j) \) as the last transformer layer’s encoding of the mask tokens of two prompts \( x_i \) and \( x_j \), respectively. Then we perform linear interpolation to produce a virtual representation:

\[
\hat{m}_{ij} = \lambda h(x_i) + (1 - \lambda) h(x_j)
\]

where \( \lambda \sim \beta(\alpha, \alpha) \). The corresponding target labels are linearly interpolated as well by:

\[
\hat{y}_{ij} = \lambda y_i + (1 - \lambda) y_j
\]

Considering an augmented multilingual verbalizer as in Section 3.1, the training objective of this particular virtual example would be:

\[
\arg\max_{\theta} \frac{1}{|L|} \sum_{\ell \in L} \{ \lambda \log P(\langle \text{mask} \rangle = V(\ell)(y_i; \theta) | \hat{m}_{ij}; \theta) \\
+ (1 - \lambda) \log P(\langle \text{mask} \rangle = V(\ell)(y_j; \theta) | \hat{m}_{ij}; \theta) \}
\]

The interpolation is performed in a dynamic in-batch fashion. For a batch drawn from the training set, we use every two adjacent examples to generate a virtual mask token representation. For more discussion about the mixup strategy for prompting methods, please refer to Appendix. A.

### 4 Experiments

In this section, we evaluate two multilingual tasks to demonstrate the effectiveness of our two-level augmentation framework.

#### 4.1 Setup

**Datasets** We conduct experiments on two sentence-pair classification tasks: XNLI (Conneau et al., 2018; Williams et al., 2018) for cross-lingual natural language inference and PAWS-X (Yang et al., 2019) for multilingual paraphrase identification. For these two datasets, while the evaluation data is human-translated, the golden training data is only available in English.

**Evaluation** Following Zhao and Schütze (2021), we conduct our experiments by training the XLM-R base model (Conneau et al., 2020) on English. Then the model will be directly applied to other target languages, without using any training examples of the target language. To make a reasonable comparison between finetuning and prompting, we ensure finetuning to be better than a random guess on each language. Therefore, we randomly sample without replacement \( K \in \{16, 32, 64, 128, 256\} \) per class for XNLI and \( K \in \{256, 512\} \) per class for PAWS-X to construct the training set. Then we use the same number of shots from the development split to perform model selection to simulate a realistic few-shot setting (Perez et al., 2021).

The evaluation of few-shot cross-lingual transfer could be with large variance and depend on the selection of few shots (Zhang et al., 2021; Zhao et al., 2021; Keung et al., 2020). In our work, to faithfully reflect the performance of few-shot learning, we do not follow Zhao and Schütze (2021) to fix the training/development data but randomly sample separate training/development sets for different runs.

#### 4.2 Results

Table 3 and 4 presents the accuracy on XNLI and PAWS-X dataset, respectively.

**UP v.s. Finetuning** On the XNLI dataset, even the simplest prompting method for cross-lingual transfer, namely UP, consistently outperforms the
Table 4 also exhibits consistent superiority of our method over cross-lingual finetuning. Even in the most resource-rich settings, compared to FT, our method still obtains 7.1% (256 shots) and 4.9% (512 shots) absolute gains on XNLI and PAWS-X.

Ablation Study Our performance of the proposed method will become worse when we remove representation-level mixup or multilingual verbalizer, showing that both of the augmentation strategies defined at representation-level and answer-level contribute positively to the improvement. We also notice that the negative effects brought by OURS w/o MV are generally larger, showing that the guidance from multiple target languages is more helpful for cross-lingual prompting.

Inference Strategy Our augmentation framework can be naturally extended by designing more sophisticated inference strategies. Interestingly, we find that English-only inference is still comparable to these strategies. More discussions can be found in Appendix C.

5 Conclusion

In this paper, we first derive Universal Prompting, a simple but effective baseline for cross-lingual prompting. The proposed two-level augmentation framework further enhance cross-lingual prompting on two sentence-pair classification tasks. In the future, we will consider verifying the effectiveness of prompting and the proposed augmentation framework in cross-lingual sequence tagging or text generation tasks.
References


A Additional Implementation Details

Implementation Package Our implementation is based on PyTorch (Paszke et al., 2019) and Huggingface Transformer (Wolf et al., 2019) framework.

Model Details XLM-R base model, containing 270M parameters, is pretrained on 2.5TB of filtered CommonCrawl on 100 languages. It contains 12 Transformer layers with hidden space dimensions of 768 and 12 attention heads in each layer.

Computing Infrastructure All of our experiments are conducted on a single Tesla V100-SXM2 32G. Gradient accumulation steps of 4 is used for prompting to overcome resource limitations.

Hyperparameter Settings Our major hyperparameter settings follow Zhao and Schütze (2021). A fixed learning rate (1e-5) is used for all of our experiments without any learning rate schedule to compare finetuning with prompting (Le Scao and Rush, 2021). We use a smaller batch size of 8 for finetuning and prompting because it achieves slightly better performance. We use the max sequence length of 256. The model is trained for 50 epochs and we select the checkpoint by development accuracy for testing as suggested in Mosbach et al. (2021); Zhang et al. (2021). The α value for β distribution in representation-level mixup is set to 1.2 for all of the experiments.

Promting The language sets ℒ used for multilingual verbalizers are determined by the language availability of the dataset. Specifically, for XNLI, ℒ = {EN, AR, BG, DE, EL, ES, FR, HI, RU, SW, TH, TR, UR, VI, ZH}. For PAWS-X, ℒ = {EN, DE, ES, FR, JA, KO, ZH}

For simplicity, the verbalizers of target languages are translated by Google Translate. Similar with XNLI, we use "paraphrase → yes" and "non-paraphrase → no" as the verbalizer of PAWS-X in English. Table 5 presents the full multilingual verbalizer we use for the PAWS-X dataset.

We discuss Universal Promtting across languages for multilingual sentence-pair classification tasks in Section 2. Moreover, we believe the same notion of alleviating source-target discrepancies in terms of prompt template and verbalizer is also applicable for cross-lingual single-sentence classification or text generation tasks, which is left for future work.

<table>
<thead>
<tr>
<th>Language</th>
<th>Verbalizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN</td>
<td>Paraphrase → yes</td>
</tr>
<tr>
<td></td>
<td>Non-paraphrase → no</td>
</tr>
<tr>
<td>DE</td>
<td>Paraphrase → Ja</td>
</tr>
<tr>
<td></td>
<td>Non-paraphrase → Nein</td>
</tr>
<tr>
<td>ES</td>
<td>Paraphrase → si</td>
</tr>
<tr>
<td></td>
<td>Non-paraphrase → no</td>
</tr>
<tr>
<td>FR</td>
<td>Paraphrase → Oui</td>
</tr>
<tr>
<td></td>
<td>Non-paraphrase → non</td>
</tr>
<tr>
<td>JA</td>
<td>Paraphrase → はい</td>
</tr>
<tr>
<td></td>
<td>Non-paraphrase → ない</td>
</tr>
<tr>
<td>ZH</td>
<td>Paraphrase → 是</td>
</tr>
<tr>
<td></td>
<td>Non-paraphrase → 否</td>
</tr>
<tr>
<td>KO</td>
<td>Paraphrase → 예</td>
</tr>
<tr>
<td></td>
<td>Non-paraphrase → 아니</td>
</tr>
</tbody>
</table>

Table 5: The multilingual verbalizer for PAWS-X.

B Additional Discussion about Mixup Strategy for Promting

Previous mixup methods for NLP models perform the interpolation at the input embedding level (Zhang and Vaidya, 2021), hidden representation level (Jindal et al., 2020; Chen et al., 2020) or the [CLS] token (Zhang and Vaidya, 2021). However, none of them is directly applicable for prompting-based methods. In prompting-based methods, the most important hidden space representation for classification is encoded at the position of mask tokens. Different training data may have different sequence lengths and their mask tokens may be put at different positions. Previous practices of hidden representation level mixup will result in the interpolation between the representation of a mask token and a normal token, which is meaningless in prompting methods. Therefore, we find that the most intuitive way is to apply the interpolation in the last transformer layer’s representations of mask tokens. Then the interpolated representation is fed into the masked language modeling head.

Note that our proposed representation-level mixup of mask tokens is also directly applicable for monolingual prompting. It would also be interesting to apply it to more settings, which is left for future study.

C Inference Strategy

A natural extension for our method is to leverage the multilingual verbalizer in some way for target-language inference as well. For comparisons, we heuristically devise the following inference strategies:

(1) English Verbalizer The English verbalizer is still used when transferring to target languages.
This strategy is used to produce results in Table 3 and 4. To formalize:
\[
\hat{y} = \arg \max_y P(\langle \text{mask} \rangle = V_{EN}(y)|x; \theta) \tag{5}
\]

(2) Target Language Verbalizer The verbalizer in the corresponding target language is used, which is the practice of Zhao and Schütze (2021). To formalize:
\[
\hat{y} = \arg \max_y P(\langle \text{mask} \rangle = V_{target}(y)|x; \theta) \tag{6}
\]

(3) Taking Maximum over the Multilingual Verbalizer In this strategy, we will take the maximum probability over the whole multilingual verbalizer. To formalize:
\[
\hat{y} = \arg \max_{y, \ell} P(\langle \text{mask} \rangle = V_{\ell}(y)|x; \theta) \tag{7}
\]

(4) Taking Sum over the Multilingual Verbalizer In this strategy, we will take the sum of probability over the whole multilingual verbalizer. To formalize:
\[
\hat{y} = \arg \max_y \sum_{\ell \in \mathcal{L}} P(\langle \text{mask} \rangle = V_{\ell}(y)|x; \theta) \tag{8}
\]

(5) Bilingual Verbalizer In this strategy, we will take the sum of probability over the target language verbalizer and the English verbalizer. To formalize, the predicted label \( \hat{y} \) is given by:
\[
\hat{y} = \arg \max_y \left\{ P(\langle \text{mask} \rangle = V_{EN}(y)|x; \theta) + P(\langle \text{mask} \rangle = V_{target}(y)|x; \theta) \right\} \tag{9}
\]

We use the checkpoint of XLM-R trained by 128 shots on the XNLI dataset and make inference with different strategies. Table 6 shows the accuracy by employing different inference strategies. We show that with our two-level augmentation framework, the inference is quite robust to the utilization of the verbalizer. This can probably be attributed to answer-level multilingual verbalizers, which help to model label tokens in multiple languages. We choose to simply employ English-only inference due to its simplicity and slightly better performance to produce results in Tables 3 and 4.

<table>
<thead>
<tr>
<th>Strategy Num</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>56.42 ± 0.37</td>
</tr>
<tr>
<td>2</td>
<td>56.31 ± 0.15</td>
</tr>
<tr>
<td>3</td>
<td>56.23 ± 0.09</td>
</tr>
<tr>
<td>4</td>
<td>56.33 ± 0.11</td>
</tr>
<tr>
<td>5</td>
<td>56.39 ± 0.21</td>
</tr>
</tbody>
</table>

Table 6: Test accuracy by using different inference strategies. The accuracy is averaged by 15 testing languages of XNLI of 5 random seeds.