Multilingual Relation Classification via Efficient and Effective Prompting

Yuxuan Chen  David Harbecke  Leonhard Hennig
German Research Center for Artificial Intelligence (DFKI)
Speech and Language Technology Lab
{yuxuan.chen, david.harbecke, leonhard.hennig}@dfki.de

Abstract
Prompting pre-trained language models has achieved impressive performance on various NLP tasks, especially in low data regimes. Despite the success of prompting in monolingual settings, applying prompt-based methods in multilingual scenarios has been limited to a narrow set of tasks, due to the high cost of handcrafting multilingual prompts. In this paper, we present the first work on prompt-based multilingual relation classification (RC), by introducing an efficient and effective method that constructs prompts from relation triples and involves only minimal translation for the class labels. We evaluate its performance in fully supervised, few-shot and zero-shot scenarios, and analyze its effectiveness across 14 languages, prompt variants, and English-task training in cross-lingual settings. We find that in both fully supervised and few-shot scenarios, our prompt method beats competitive baselines: fine-tuning XLM-R and null prompts. It also outperforms the random baseline by a large margin in zero-shot experiments. Our method requires little in-language knowledge and can be used as a strong baseline for similar multilingual classification tasks.

1 Introduction
Relation classification (RC) is a crucial task in information extraction (IE), aiming to identify the relation between entities in a text (Alt et al., 2019). Extending RC to multilingual settings has recently received increased interest (Zou et al., 2018; Kolluru et al., 2022), but the majority of prior work still focuses on English (Baldini Soares et al., 2019; Lyu and Chen, 2021). A main bottleneck for multilingual RC is the lack of supervised resources, comparable in size to large English datasets (Riedel et al., 2010; Zhang et al., 2017). The SMiLER dataset (Seganti et al., 2021) provides a starting point to test fully supervised and more efficient approaches due to different resource availability for different languages.

Previous studies have shown the promising performance of prompting PLMs compared to the data-hungry fine-tuning, especially in low-resource scenarios (Gao et al., 2021; Le Scao and Rush, 2021; Lu et al., 2022). Multilingual pre-trained language models (Conneau et al., 2020; Xue et al., 2021) further enable multiple languages to be represented in a shared semantic space, thus making prompting in multilingual scenarios feasible. However, the study of prompting for multilingual tasks so far remains limited to a small range of tasks such as text classification (Winata et al., 2021) and natural language inference (Lin et al., 2022). To our knowledge, the effectiveness of prompt-based methods for multilingual RC is still unexplored.

To analyse this gap, we pose two research questions for multilingual RC with prompts:

**RQ1.** What is the most effective way to prompt? We investigate whether prompting should be done in English or the target language and whether to use soft prompt tokens.

**RQ2.** How well do prompts perform in different data regimes and languages? We investigate the effectiveness of our prompting approach in three scenarios: fully supervised, few-shot and zero-shot. We explore to what extent the results are related to the available language resources.

We present an efficient and effective prompt method for multilingual RC (see Figure 1) that derives prompts from relation triplets (see Section 3.1). The derived prompts include the original sentence and entities and are supposed to be filled with the relation label. We evaluate the prompts with three variants, two of which require no translation, and one of which requires minimal translation, i.e., of the relation labels only. We find that our method outperforms fine-tuning and a strong task-agnostic prompt baseline in fully supervised and few-shot scenarios, especially for relatively low-
resource languages. Our method also improves over the random baseline in zero-shot settings, and achieves promising cross-lingual performance. The main contributions of this work hence are:

- We propose a simple but efficient prompt method for multilingual RC, which is, to the best of our knowledge, the first work to apply prompt-based methods to multilingual RC (Section 3).
- We evaluate our method on the largest multi-lingual RC dataset, SMiLER (Seganti et al., 2021), and compare our method with strong baselines in all three scenarios. We also investigate the effects of different prompt variants, including insertion of soft tokens, prompt language, and the word order of prompting (Sections 4 & 5).

2 Preliminaries

We first give a formal definition of the relation classification task, and then introduce fine-tuning and prompting paradigms to perform RC.

2.1 Relation Classification Task Definition

Relation classification is the task of classifying the relationship such as date_of_birth, founded_by or parents between pairs of entities in a given context. Formally, given a relation set $\mathcal{R}$ and a text $x = [x_1, x_2, \ldots, x_n]$ (where $x_1, \ldots, x_n$ are tokens) with two disjoint spans $e_h$ and $e_t$ denoting the head and tail entity, RC aims to predict the relation $r \in \mathcal{R}$ between $e_h$ and $e_t$, or give a no_relation prediction if no relation in $\mathcal{R}$ holds.

RC is a multilingual task if the token sequences come from different languages.

2.2 Fine-tuning for Relation Classification

In fine-tuning, a task-specific linear classifier is added on top of the PLM. Fine-tuning hence introduces a different scenario from pre-training, since language model (LM) pre-training is usually formalized as a cloze-style task to predict target tokens at [MASK] (Devlin et al., 2019; Liu et al., 2019) or a corrupted span (Raffel et al., 2020; Lewis et al., 2020). For the RC task, the classifier aims to predict the target class $r$ at [CLS] or at the entity spans denoted by MARKER (Baldini Soares et al., 2019).

2.3 Prompting for Relation Classification

Prompting is proposed to bridge the gap between pre-training and fine-tuning (Liu et al., 2022; Gu et al., 2022). The essence of prompting is, by appending extra text to the original text according to a task-specific template $T(\cdot)$, to reformulate the downstream task to an LM pre-training task such as masked language modeling (MLM), and apply the same training objective during the task-specific training. For the RC task, to identify the relation between “Angela Merkel” and “Joachim Sauer” in the text “Angela Merkel’s current husband is quantum chemist Joachim Sauer,” an intuitive template for prompting can be “The relation between Angela Merkel and Joachim Sauer is [MASK],” and the LM is supposed to assign a higher likelihood to the term couple than to e.g. friends or colleagues at
We introduce its template and verbalizer, and propose several variants of the prompt. For each prompt or its variant, we list (1) the prompt input and the target; (2) an example based on the plain text in German “Goethe schrieb Faust.” [\(v_1\):] learnable soft tokens. \(\phi^{EN}(r)\): the original (English) relation verbalization. \(\phi^L(r)\): the translated relation verbalization into the target language \(L\).

### 3 Methods

We now present our method, as shown in Figure 1. We introduce its template and verbalizer, and propose several variants of the prompt. Lastly, we explain the training and inference process.

#### 3.1 Template

For prompting (Liu et al., 2022), a prompt often consists of a template \(T(\cdot)\) and a verbalizer \(V\). Given a plain text \(x\), the template \(T\) adds task-related instruction to \(x\) to yield the prompt input

\[
x_{\text{prompt}} = T(x).
\]

Following Chen et al. (2022) and Han et al. (2021), we treat relations as predicates and use the cloze “\(e_h \ [\text{relation}] \ e_t\)” for the LM to fill in. Our template is formulated as

\[
T(x) := \text{“} x \cdot e_h \ ____ \ e_t \text{”}.
\]

In the template \(T(x)\), \(x\) is the original text and the two entities \(e_h\) and \(e_t\) come from \(x\). Therefore, our template does not introduce extra tokens, thus involves no translation at all.

#### 3.2 Verbalizer

After being prompted by \(x_{\text{prompt}}\), the PLM \(M\) predicts the masked text \(y\) at the blank. To complete an NLP classification task, a verbalizer \(\phi\) is required to bridge the set of labels \(\mathcal{Y}\) and the set of predicted texts (verbalizations \(\mathcal{V}\)). For the simplicity of our prompt, we use the one-to-one verbalizer:

\[
\phi : \mathcal{Y} \rightarrow \mathcal{V}, \ r \mapsto \phi(r).
\]

where \(r\) is a relation, and \(\phi(r)\) is the simple verbalization of \(r\). \(\phi(\cdot)\) normally only involves splitting \(r\) by “.” or “\_” and replacing abbreviations such as org with organization. E.g., the relation org-has-member corresponds to the verbalization “organization has member”. Then the prediction is formalized as

\[
p(r\vert x) \propto p(y = \phi(r)\vert x_{\text{prompt}}; \theta_M),
\]

where \(\theta_M\) denotes the parameters of model \(M\).

#### 3.3 Variants

To find the optimal way to prompt, we investigate three variants as follows.

- **Hard prompt vs soft prompt (SP)**
  - Hard prompts (a.k.a. discrete prompts) (Liu et al., 2022) are entirely formulated in natural language. Soft prompts (a.k.a. continuous prompts) consist of learnable tokens (Lester et al., 2021) that are not contained in the PLM vocabulary. Following Han et al. (2021), we insert soft tokens before entities and blanks as shown for SP in Table 1.

- **Code-switch (CS) vs in-language (IL)**
  - Relation labels are in English across almost all RC datasets. Given a text from a non-English input \(L\) with a blank, the recovered text is code-mixed after being completed with an English verbalization, corresponding to code-switch prompting. It is probably more reasonable for the PLM to fill in the blank in language \(L\). Inspired by Lin et al. (2022) and Zhao and Schütze (2021), we machine-translate the English verbalizers into the other languages.\(^1\) Table 1 visualizes both code-switch (CS)

<table>
<thead>
<tr>
<th>Prompt/Variant</th>
<th>Input</th>
<th>Target</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>null prompts</td>
<td>(x)</td>
<td>(\phi^{EN}(r))</td>
<td>Goethe schrieb Faust. ____</td>
</tr>
<tr>
<td>CS</td>
<td>(x, e_h \ ____ \ e_t)</td>
<td>(\phi^{EN}(r))</td>
<td>Goethe schrieb Faust. Faust ____ Goethe has author</td>
</tr>
<tr>
<td>SP</td>
<td>(x, [v_1], e_h \ [v_1] \ [v_2] e_t)</td>
<td>(\phi^{EN}(r))</td>
<td>Goethe schrieb Faust. [v_1]Faust [v_2] ____ [v_3]Goethe has author</td>
</tr>
<tr>
<td>IL</td>
<td>(x, e_h \ ____ \ e_t)</td>
<td>(\phi^L(r))</td>
<td>Goethe schrieb Faust. Faust ____ Goethe has Autor</td>
</tr>
</tbody>
</table>

1See Appendix B for more examples of translated verbalizations. To translate the verbalizer of the SMiLER dataset, we use DeepL by default and Google Translate when the target language is not supported by DeepL (in case of AR, FA, KO and UK).
Table 2: Statistics of the lengths of the verbalizations over several classification tasks. The lengths for non-RC tasks depend on the tokenizers from the respective PLMs in the cited work. The lengths for RC tasks are based on the mT5\textsubscript{BASE} tokenizer. Mean and std. show that the label space of the RC task is more complex than most few-class classification tasks. The verbalizations of RC datasets are listed in Appendix B. For SemEval, the two possible directions of a relation are combined. For NYT, we use the version from Zeng et al. (2018). For SMiLER, “EN” is the English split; “ALL” contains all data from 14 languages.

<table>
<thead>
<tr>
<th>Task</th>
<th>Dataset</th>
<th>#Class</th>
<th>Verbalizations</th>
<th># Token in Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>LA</td>
<td>ColA (Wartstadt et al., 2019)</td>
<td>2</td>
<td>correct, incorrect. (Gao et al., 2021)</td>
<td>1</td>
</tr>
<tr>
<td>NER</td>
<td>CoNLL03 (Tjong Kim Sang and De Meulder, 2003)</td>
<td>5</td>
<td>location, person, not an, … (Cui et al., 2021)</td>
<td>1.2</td>
</tr>
<tr>
<td>NLI</td>
<td>MNLI (Williams et al., 2018)</td>
<td>3</td>
<td>yes, no, maybe. (Fu et al., 2022)</td>
<td>1</td>
</tr>
<tr>
<td>NLI</td>
<td>XNLI (Conneau et al., 2018)</td>
<td>3</td>
<td>yes, no, maybe; Evet, … (Zhao and Schütze, 2021)</td>
<td>1</td>
</tr>
<tr>
<td>PI</td>
<td>PAWS-X (Yang et al., 2019)</td>
<td>2</td>
<td>yes, no. (Qi et al., 2022)</td>
<td>1</td>
</tr>
<tr>
<td>TC</td>
<td>MARC (Keung et al., 2020)</td>
<td>2</td>
<td>good, (average, bad). (Huang et al., 2022)</td>
<td>1</td>
</tr>
</tbody>
</table>

and in-language (IL) prompting. For English, CS- and IL- prompting are equivalent, since $L$ is English itself.

**Word order of prompting** For the RC task, head-relation-tail triples involve three elements. Therefore, deriving natural language prompts from them requires handling where to put the predicate (relation). In the case of SOV languages, filling in a relation that occurs between $e_h$ and $e_t$ seems less intuitive. Therefore, to investigate if the word order of prompting affects prediction accuracy, we swap the entities and the blank in the SVO-template “$x$. $e_h$ _____ $e_t$” and get “$x$. $e_t$ _____ $e_h$” as the SOV-template.

### 3.4 Training and Inference

The training and inference setups depend on the employed model. Prompting autoencoding language models requires the verbalizations to be of fixed length, since the length of masks, which is identical with verbalization length, is unknown during inference. Encoder-decoders can handle verbalizations of varying length by nature (Han et al., 2022; Du et al., 2022). Han et al. (2021) adjust all the verbalizations in TACRED to a length of 3, to enable prompting with RoBERTa for RC. We argue that for multilingual RC, this fix is largely infeasible, because: (1) in case of in-language prompting on SMiLER, the variance of the length of the verbalizations increases from 0.68 to 1.44 after translation (see Table 2), and surpasses most of listed monolingual RC datasets (SemEval, NYT and SciERC), making it harder to unify the length; (2) manually adjusting the translated prompts requires manual effort per target language, making it much more expensive than adjusting only English verbalizations. Therefore, we suggest using an encoder-decoder PLM for prompting (Song et al., 2022).

**Training objective** For an encoder-decoder PLM $M$, given the prompt input $T(x)$ and the target sequence $\phi(r)$ (i.e. label verbalization), we denote the output sequence as $y$. The probability of an exact-match decoding is calculated as follows:

$$
\prod_{t=1}^{|\phi(r)|} P_\theta(y_t = \phi_t(r)|y_{<t}, T(x)) ,
$$

where $y_t$, $\phi_t(r)$ denote the $t$-th token of $y$ and $\phi(r)$, respectively. $y_{<t}$ denotes the decoded sequence on the left. $\theta$ represents the set of all the learnable parameters, including those of the PLM $\theta_M$, and those of the soft tokens $\theta_{vp}$ in case of variant “soft prompt”. Hence, the final objective over the training set $X$ is to minimize the negative log-likelihood:

$$
\arg \min_{\theta} - \frac{1}{|X|} \sum_{x \in X} \sum_{t=1}^{|\phi(r)|} P_\theta(y_t = \phi_t(r)|y_{<t}, T(x)) .
$$

**Inference** We collect the output logits of the decoder, $L \in \mathbb{R}^{|V| \times L}$, where $|V|$ is the vocabulary size of $M$, and $L$ is the maximum decode length. For each relation $r \in R$, its score is given by (Han et al., 2022):

$$
score_\theta(r) := \frac{1}{|\phi(r)|} \sum_{t=1}^{|\phi(r)|} P_\theta(y_t = \phi_t(r)) ,
$$
<table>
<thead>
<tr>
<th>Lang.</th>
<th>#Class</th>
<th>#Train(K)</th>
<th>Max.</th>
<th>mT5(B)</th>
<th>XLM-R(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>9</td>
<td>9.3</td>
<td>74</td>
<td>57</td>
<td>2.9</td>
</tr>
<tr>
<td>DE</td>
<td>22</td>
<td>51.5</td>
<td>84</td>
<td>347</td>
<td>10.3</td>
</tr>
<tr>
<td>EN</td>
<td>36</td>
<td>267.6</td>
<td>110</td>
<td>2733</td>
<td>55.6</td>
</tr>
<tr>
<td>ES</td>
<td>21</td>
<td>11.1</td>
<td>70</td>
<td>433</td>
<td>9.4</td>
</tr>
<tr>
<td>FA</td>
<td>8</td>
<td>2.6</td>
<td>93</td>
<td>52</td>
<td>13.3</td>
</tr>
<tr>
<td>FR</td>
<td>22</td>
<td>60.9</td>
<td>83</td>
<td>318</td>
<td>9.8</td>
</tr>
<tr>
<td>IT</td>
<td>22</td>
<td>74.0</td>
<td>86</td>
<td>162</td>
<td>5.0</td>
</tr>
<tr>
<td>KO</td>
<td>28</td>
<td>18.7</td>
<td>95</td>
<td>26</td>
<td>5.6</td>
</tr>
<tr>
<td>NL</td>
<td>22</td>
<td>38.9</td>
<td>76</td>
<td>73</td>
<td>5.0</td>
</tr>
<tr>
<td>PL</td>
<td>21</td>
<td>16.8</td>
<td>86</td>
<td>130</td>
<td>6.5</td>
</tr>
<tr>
<td>PT</td>
<td>22</td>
<td>43.3</td>
<td>82</td>
<td>146</td>
<td>8.4</td>
</tr>
<tr>
<td>RU</td>
<td>8</td>
<td>6.4</td>
<td>69</td>
<td>713</td>
<td>23.4</td>
</tr>
<tr>
<td>SV</td>
<td>22</td>
<td>4.5</td>
<td>84</td>
<td>45</td>
<td>0.08</td>
</tr>
<tr>
<td>UK</td>
<td>7</td>
<td>1.0</td>
<td>65</td>
<td>41</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Table 3: Statistics of the 14 languages in the SMiLER dataset, including the number of classes, the number of training examples (in thousands), and the maximum text length over train and test splits. Appended to the table are the sizes (in billion tokens) of pre-training corpora of the referred languages for mT5 and XLM-R, respectively.

where we compute $P$ by looking up in the $t$-th column of $L$ and applying softmax at each time step $t$. We aggregate $P$ by addition to encourage partial matches as well, instead of enforcing exact matches. The score is normalized by the length of verbalization in order to avoid predictions favoring longer relations. Finally, we select the relation with the highest score as prediction.

4 Experiments

We implement our experiments using the Hugging Face Transformers library (Wolf et al., 2020), Hydra (Yadan, 2019) and PyTorch (Paszke et al., 2019). We use micro-F1 as the evaluation metric, as the SMiLER paper (Seganti et al., 2021) suggests. To measure the overall performance over multiple languages, we report the macro average across languages, following Zhao and Schütze (2021) and Lin et al. (2022). We also group the languages by their available resources in both pre-training and fine-tuning datasets for additional aggregate results. Details of the dataset, the models, and the experimental setups are as follows. Further experimental details are listed in Appendix A.

4.1 Dataset

We conduct an experimental evaluation of our multilingual prompt methods on the SMiLER (Seganti et al., 2021) dataset, which contains 1.1M annotated texts across 14 languages. Table 3 lists the main statistics of the different languages in the SMiLER dataset. Note that languages have varying number of relations, mostly related to how many samples are present. We do not evaluate other datasets because the only prior multilingual RC dataset that fits our task, RELX (Köksal and Özgür, 2020), contains only 502 parallel examples in 5 languages.

Grouping of the languages We visualize the languages in Figure 2 based on the sizes of RC training data, but include the pre-training data as well, to give a more comprehensive overview of the availability of resources for each language. We divide the 14 languages into 4 groups, according to the detectable clusters in Figure 2 and language origins.

4.2 Model

For prompting, we use mT5$_{\text{BASE}}$ (Xue et al., 2021), an encoder-decoder PLM that supports 101 languages, including all languages in SMiLER. mT5$_{\text{BASE}}$ (Xue et al., 2021) has 220M parameters.

4.3 Baselines

EN(B) (Seganti et al., 2021) EN(B) is the baseline proposed together with the SMiLER dataset.

Note that SMiLER contains 3 versions of the English split: en (268K training examples), en-small (36K) and en-full (744K). We use the en version by default, unless specified otherwise.
They fine-tune BERT$_{\text{BASE}}$ on the English training split and report the micro-F1 on the English test split. BERT$_{\text{BASE}}$ has 110M parameters.

**XLM-R$^3$** To provide a fine-tuning baseline, we re-implement BERT$_{\text{EM}}$ (Baldini Soares et al., 2019) with the ENTITY START variant. In this method, the top-layer representations at the starts of the two entities are concatenated for linear classification. To adapt BERT$_{\text{EM}}$ to multilingual tasks, we change the PLM from BERT to a multilingual autoencoder, XLM-R$_{\text{BASE}}$ (Conneau et al., 2020), and refer to this model as XLM-R$_{\text{EM}}$. XLM-R$_{\text{BASE}}$ has 125M parameters.

**Null prompts** (Logan IV et al., 2022) To better verify the effectiveness of our method, we implement null prompts as a strong task-agnostic baseline. Null prompts involve minimal prompt engineering by directly asking the LM about the relation, without giving any task instruction (see Table 1). Logan IV et al. (2022) show that null prompts surprisingly achieve on-par performance with handcrafted prompts on many tasks. For best comparability, we use the same PLM mT5$_{\text{BASE}}$.

### 4.4 Fully Supervised Setup

We evaluate the performance of XLM-R$_{\text{EM}}$, null prompts, and our method on each of the 14 languages, after training on the full train split from that language. The prompt input and target of null prompts and our prompts are listed in Table 1.

We employ the randomly generated seed 319 for all the evaluated methods. For XLM-R$_{\text{EM}}$, we follow Baldini Soares et al. (2019) and set the batch size to be 64, the optimizer to be Adam with the learning rate $3 \times 10^{-5}$ and the number of epochs to be 5. For null prompts and ours, we use AdamW as the optimizer with the learning rate $3 \times 10^{-5}$, as Zhang et al. (2022) suggest for most of the sequence-to-sequence tasks, the number of epochs to 5, and batch size to 16. The maximum sequence length is 256 for all methods.

### 4.5 Few-shot Setup

Few-shot learning is normally cast as a $K$-shot problem, where $K$ labelled examples per class are available. We follow Chen et al. (2022) and Han et al. (2021), and evaluate on 8, 16 and 32 shots.

The few-shot training set $D_{\text{train}}$ is generated by randomly sampling $K$ instances per relation from the training split. The test set $D_{\text{test}}$ is the original test split from that language. We follow Gao et al. (2021) and sample another $K$-shot set from the English train split as validation set $D_{\text{val}}$. We tune hyperparameters on $D_{\text{val}}$ for the English task, and apply these to all languages.

We evaluate the same methods as in the fully supervised scenarios, but repeat 5 runs as suggested in Gao et al. (2021), and report the mean and standard deviation of micro-F1. We use a fixed set of random seeds $\{13, 36, 121, 223, 319\}$ for data generation and training across the 5 runs. For XLM-R$_{\text{EM}}$, we use the same hyperparameters as Baldini Soares et al. (2019), a batch size of 256, and a learning rate of $1 \times 10^{-4}$. For null prompts and our prompts, we set the learning rate to $3 \times 10^{-4}$, batch size to 16, and the number of epochs to 20.

### 4.6 Zero-shot Setup

We consider two scenarios for zero-shot multilingual relation classification.

**Zero-shot in-context learning** Following Kojima et al. (2022), we investigate whether PLMs are also decent zero-shot reasoners for RC. This scenario does not require any samples or training. We test the out-of-the-box performance of the PLM by directly prompting it with $x_{\text{prompt}}$. Zero-shot in-context learning does not specify further hyperparameters since it is training-free.

**Zero-shot cross-lingual transfer** In this scenario, following Krishnan et al. (2021), we fine-tune the model with in-language prompting on the English train split, and then conduct zero-shot in-context tests with this fine-tuned model on other languages using code-switch prompting. Through this setting, we want to verify if task-specific pre-training in a high-resource language such as English helps in other languages. In zero-shot cross-lingual transfer, we use the same hyperparameters and random seed to fine-tune on the English task.

### 5 Results and Discussion

We first present the results in fully supervised, few-shot and zero-shot scenarios, and then discuss the main findings for answering the research questions in Section 1.

#### 5.1 Fully Supervised Results

Table 4 presents the experimental results in the fully supervised scenario, for different methods, languages, and language groups. We see that all
We conclude that, with sufficient supervision, in-language prompting performs best, especially for lower-resource languages. Table 5 presents the per-group results in few-shot scenarios. We consider the random baseline for comparison (Zhao and Schütze, 2021; Winata et al., 2021). We notice that performance of the random baseline varies a lot across languages, since the dataset (cf. Table 3), with English being the hardest task. The other two variants, code-switch prompting with and w/o soft tokens, achieve $F_1$ scores of 84.1 and 82.7, respectively, only 0.9 and 2.3 lower than in-language. All three prompt variants are hence effective in fully supervised scenarios.

On a per-group basis, we find that the lower-resourced a language is, the greater an advantage prompting enjoys against fine-tuning. In particular, in-language prompts shows better robustness compared to XLM-R$_{\text{EM}}$ in low-resource languages. They both yield 95.9-96.0 $F_1$ scores for English, but XLM-R$_{\text{EM}}$ decreases to 54.3 and 3.7 $F_1$ in Group-M and -L, while in-language prompting still delivers 83.5 and 65.2 $F_1$.

### 5.2 Few-shot Results

Table 5 presents the per-group results in few-shot experiments. All the methods benefit from larger $K$. Similarly, in-language prompting still turns out to be the best contender, performing 1st in 8- and 32-shot, and the 2nd in 16-shot. We see that in-language outperforms XLM-R$_{\text{EM}}$ in all $K$-shots, while code-switch achieves comparable or even lower $F_1$ to XLM-R$_{\text{EM}}$ for $K = 8$, suggesting that the choice of prompt affects the few-shot performance greatly, thus needs careful consideration.

On a per-group basis, we find that in-language prompting outperforms other methods for middle- and low-resourced languages. Similar observations can also be drawn from fully supervised results. We conclude that, with sufficient supervision, in-language is the optimal variant to prompt rather than code-switch. We hypothesize it is due to the pre-training scenario, where the PLM rarely sees code-mixed text (Santy et al., 2021).

### 5.3 Zero-shot Results

Table 6 presents the per-language results in zero-shot scenarios. We consider the random baseline for comparison (Zhao and Schütze, 2021; Winata et al., 2021). We notice that performance of the random baseline varies a lot across languages, since the languages have different number of classes in the dataset (cf. Table 3), with English being the hardest task.

For zero-shot in-context, code-switch prompting always outperforms the random baseline by a large
Table 6: Zero-shot results in micro-F1 (%) on the SMiLER dataset. "SVO" and "SOV": word order of prompting. Overall, Code-switch prompting performs the best in the zero-shot in-context scenario. In cross-lingual transfer experiments, English-task training greatly improves the performance on all the other 13 languages.

5.4 Discussion

Based on the results above, we answer the research questions from Section 1.

RQ1. Which is the most effective way to prompt? In the fully-supervised and few-shot scenario, in-language prompting displays the best results. This appears to stem from a solid performance across all languages in both settings. Its worst performance is 31.8 $F_1$ for Polish 8-shot (see Table 7 in Appendix C). All other methods have results lower than 15.0 $F_1$ for some language. This indicates that with little supervision mT5 is able to perform the task when prompted in the language of the original text. However, zero-shot results strongly prefer code-switch prompting. It could follow that, without fine-tuning, the model’s understanding of this task is much better in English.

RQ2. How well does our method perform in different data regimes and languages? Averaged over all languages, all our variants outperform the baselines, except for 8-shot. For some high-resource languages, XLM-R $\text{EM}$ is able to outperform our method. On the other hand, for low-resource languages null prompts are a better baseline which we consistently outperform. This could indicate that prompting the underlying mT5 model is better suited for multilingual RC on SMiLER. Overall, the results suggest that minimal translation can be very helpful for multilingual relation classification.

6 Related Work

Multilingual relation classification

Previous work in multilingual RC has primarily focused on traditional methods rather than prompting PLMs. Faruqui and Kumar (2015) machine-translate non-English full text to English to deal with multilinguality. Akbik et al. (2016) employ a shared semantic role labeler to get language-agnostic abstraction and apply rule-based methods to classify the unified abstractions. Lin et al. (2017) employ convolutional networks to extract relation embeddings from texts, and propose cross-lingual attention between
relation embeddings to model cross-lingual information consistency. Sanh et al. (2019) leverage the embeddings from BiLSTM, which is trained with a set of selected semantic tasks to help (multilingual) relation extraction. Köksal and Özgür (2020) fine-tune (multilingual) BERT, classifying the embedding at [CLS]. To take entity-related embeddings into consideration as well, Nag et al. (2021) add an extra summarization layer on top of a multilingual BERT to collect and pool the embeddings at both [CLS] and entity starts.

**Multilingual prompting**

Multilingual prompting is a new yet fast-growing topic. Winata et al. (2021) reduce handcrafting efforts by reformulating general classification tasks into binary classification with answers restricted to true or false for all languages. Huang et al. (2022) propose a unified multilingual prompt by introducing a so-called “two-tower” encoder, with the template tower producing language-agnostic prompt representation, and the context tower encoding text information. Fu et al. (2022) manually translate prompts and suggest multilingual multitask training to boost the performance for a target downstream task.

7 Conclusion

In this paper, we present a first, simple yet efficient and effective prompt method for multilingual relation classification, by translating only the relation labels. Our prompting outperforms fine-tuning and null prompts in fully supervised and few-shot experiments. With supervised data, in-language prompting enjoys the best performance, while in the zero-shot scenarios prompting in English is preferable. We attribute the good performance of our method to its well-suitedness for RC, with the derivation of $entity_1$-relation-$entity_2$ prompts from relation triples. We would like to see our method extended to similar tasks, such as semantic role labeling, with a structure between concepts that can be described in natural language.

**Limitations**

We acknowledge the main limitation of this work is that we only experiment on one dataset with 14 languages. Multilingual RC datasets prior to SMiLER are limited in the coverage of languages or in the size of unique training examples. It would be interesting to see how our method performs on other multilingual RC datasets, especially for underrepresented languages (Winata et al., 2022).

We restrict the target language to be supported by the underlying PLM. The popular multilingual PLMs, mT5 and mBART, include 101 and 25 languages during pre-training. We rely on these PLMs and fail to study true low-resource languages that are not represented in such PLMs (Aji et al., 2022).

It is noticeable that in the fully supervised scenario, for 7 out of the 14 languages, at least one method achieves over 0.95 micro-$F_1$ score. We hypothesize that is due to high homogeneity in and between the train and test split. If so, the dataset itself might not be challenging, which could indicate that the results are mostly measuring how well the model is able to fit a few indicators (quickly).

Like most other prompt methods, ours requires the label names to be natural language which are indicative of the class. Therefore, our method would suffer from labels being non-descriptive.

**Ethics Statement**

We use automated machine translation by Google Translate and DeepL for our method. These MT systems contain biases regarding, e.g., gender (“has-author”: “hat Autor”) where gender-neutral English nouns are translated to gendered nouns in target languages.

In this work we evaluate SMiLER (Seganti et al., 2021), which is crawled from Wikipedia. In the paper, they have not stated measures that prevent collecting sensitive text. Therefore, we do not rule out the possible risk of sensitive content in the data.

The PLMs involved in this paper are BERT$_{BASE}$ for EN(B), XLM-R$_{BASE}$ for XLM-R$_{EM}$, and mT5$_{BASE}$ for null prompts and ours. BERT$_{BASE}$ is pre-trained on the BooksCorpus (Zhu et al., 2015) and English Wikipedia. XLM-R$_{EM}$ is pre-trained on a CommonCrawl corpus. mT5$_{BASE}$ is pre-trained on mC4, a filtered CommonCrawl corpus. All our published models may have inherited biases from these corpora.

**Acknowledgments**

We would like to thank Nils Feldhus and the anonymous reviewers for their valuable comments and feedback on the paper. This work has been supported by the German Federal Ministry for Economic Affairs and Climate Action as part of the project PLASS (01MD19003E), and by the German Federal Ministry of Education and Research as part of the projects CORA4NLP (01IW20010) and BBDC2 (01IS18025E).
References


Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In ICML 2022 Workshop on Knowledge Retrieval and Language Models.


Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In Proceedings of the 2018


A Experimental Details

A.1 Hyperparameter Search

We investigated the following possible hyperparameters for few-shot settings. For fully-supervised, we take hyperparameters from literature (see Section 4.4).

Number of epochs: $[10, 20]$; Learning rate: $[1 \times 10^{-5}, 3 \times 10^{-5}, 1 \times 10^{-4}, 3 \times 10^{-4}]$; Batch size: $[16, 64, 256]$, not tuned but selected based on available GPU VRAM.

We manually tune these hyperparameters, based on the micro-$F_1$ score on the validation set.

A.2 Computing Infrastructure

Fully supervised experiments are conducted on a single A100-80GB GPU. Few-shot and zero-shot experiments are conducted on a single A100 GPU.

A.3 Average Running Time

Fully supervised It takes 5 hours to train for 1 run with mT5BASE and a prompt method (null prompts,
CS, SP and IL) on either English, or all other languages in total. With XLM-R_{EM} the running time is 3 hours.

**Few-shot** It takes 20 (8-shot), 26 (16-shot), and 36 minutes (32-shot) for 1 run with mT5_{BASE} and a prompt method over all languages. With XLM-R_{EM} the running time is 8 minutes.

**Zero-shot** For zero-shot in-context experiments, it takes 6 minutes with mT5_{BASE} and a prompt method over all languages. For zero-shot cross-lingual transfer, the running time equals English training time (5 hours) plus inference-only time (6 minutes).

## B Verbalizers for SMiLER

- **EN** "birth-place": "birth place", "eats": "eats", "event-year": "event year", "first-product": "first product", "from-country": "from country", "has-author": "has author", "has-child": "has child", "has-edu": "has education", "has-genre": "has genre", "has-height": "has height", "has-highest-mountain": "has highest mountain", "has-length": "has length", "has-lifespan": "has lifespan", "has-nationality": "has nationality", "has-occupation": "has occupation", "has-parent": "has parent", "has-population": "has population", "has-sibling": "has sibling", "has-spouse": "has spouse", "has-tourist-attraction": "has tourist attraction", "has-type": "has type", "has-weight": "has weight", "headquarters": "headquarters", "invented-by": "invented by", "invented-when": "invented when", "is-member-of": "is member of", "is-where": "located in", "loc-leader": "location leader", "movie-has-director": "movie has director", "movie-has-director": "movie has director", "no_relation": "no relation", "org-has-founder": "organization has founder", "org-has-member": "organization has member", "org-leader": "organization leader", "post-code": "post code", "starowing": "starring", "won-award": "won award";

- **AR** "year": "السنة", "has-edu": "لدنه تعليم", "has-genre": "لديه جنس", "has-education": "المدرسة", "has-population": "عدد السكان", "has-type": "النوع", "is-member-of": "عضو في", "no_relation": "لا علاقة", "won-award": "فاز بالجائزة";

- **DE** "birth-place": "Geburtsort", "event-year": "Veranstaltungsjahr", "from-country": "vom Land", "has-author": "hat Autor", "has-child": "hat Kind", "has-edu": "hat Bildung", "has-genre": "hat Genre", "has-occupation": "hat Beruf", "has-parent": "hat Eltern-teil", "has-population": "hat Bevölkerung", "has-spouse": "hat Ehepartner", "has-type": "hat Typ", "headquarters": "Hauptsitz", "is-member-of": "ist Mitglied von", "is-where": "gelegen in", "loc-leader": "Standortleiter", "movie-has-director": "Film hat Regisseur", "no_relation": "keine Beziehung", "org-has-founder": "Organisation hat Gründer", "org-has-member": "Organisation hat Mitglied", "org-leader": "Organisationsleiter", "won-award": "gewann eine Auszeichnung";

- **ES** "birth-place": "lugar de nacimiento", "event-year": "año de evento", "from-country": "del país", "has-author": "tiene autor", "has-child": "tiene hijo", "has-edu": "tiene educación", "has-genre": "tiene género", "has-occupation": "tiene ocupación", "has-parent": "tiene padre", "has-population": "tiene población", "has-spouse": "tiene cónyuge", "has-type": "tiene tipo", "headquarters": "sede central", "is-member-of": "es miembro de", "is-where": "situado en", "loc-leader": "líder de ubicación", "movie-has-director": "película cuenta con el director", "no_relation": "sin relación", "org-has-founder": "organización cuenta con el fundador", "org-has-member": "organización tiene miembro", "won-award": "ganó el premio";

- **FA** "year": "سال رویداد", "has-edu": "نحوه تحصیلات دارد", "has-genre": "ژانر دارد", "has-occupation": "شغل دارد", "has-population": "جمعیت دارد", "has-type": "نوع دارد", "is-member-of": "عضو است", "no_relation": "هیچ رابطه ای";

- **FR** "birth-place": "lieu de naissance", "event-year": "année de l’événement", "from-country": "du pays", "has-author": "a un auteur", "has-child": "a un enfant", "has-edu": "a une éducation", "has-genre": "a un genre", "has-occupation": "a une profession", "has-parent": "a un parent", "has-population": "a de la population", "has-spouse": "a un conjoint", "has-type": "a le type", "headquar-
"org-leader": "líder da organização", "won-award": "ganhou prémio";

• RU "event-year": "год события", "has-edu": "имеет образование", "has-genre": "имеет жанр", "has-occupation": "имеет профессию", "has-population": "имеет население", "has-type": "имеет тип", "is-member-of": "является членом", "no_relation": "без связи";

• SV "birth-place": "födelseort", "event-year": "År för evenemanget", "from-country": "från ett land", "has-author": "har en författare", "has-child": "har chilî", "has-edu": "har utbildning", "has-genre": "har en genre", "has-occupation": "har ockuperat", "has-parent": "har en förälder", "has-population": "har en befolkning", "has-spouse": "har make eller maka", "has-type": "har typ", "headquarters": "huvudkontor", "is-member-of": "är medlem i", "is-where": "som ligger i", "loc-leader": "platsansvarig", "movie-has-director": "filmen har regissör", "no_relation": "ingen relation", "org-has-founder": "organisationen har en grundare", "org-has-member": "organisationen har en medlem", "org-leader": "ledare för organisationen", "won-award": "vann ett pris";

• UK "event-year": "рiк подiї", "has-edu": "має освiту", "has-genre": "має жанр", "has-occupation": "має заняття", "has-population": "має населення", "has-type": "має тип", "no_relation": "нiякого вiдношення".
### Detailed Few-shot Results

<table>
<thead>
<tr>
<th>Shots</th>
<th>Method</th>
<th>AR</th>
<th>DE</th>
<th>EN</th>
<th>ES</th>
<th>FA</th>
<th>FR</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shots</th>
<th>Method</th>
<th>AR</th>
<th>DE</th>
<th>EN</th>
<th>ES</th>
<th>FA</th>
<th>FR</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shots</th>
<th>Method</th>
<th>AR</th>
<th>DE</th>
<th>EN</th>
<th>ES</th>
<th>FA</th>
<th>FR</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shots</th>
<th>Method</th>
<th>AR</th>
<th>DE</th>
<th>EN</th>
<th>ES</th>
<th>FA</th>
<th>FR</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
</tr>
</tbody>
</table>

Table 7: Few-shot results in micro-F1 (%) on the SMiLER dataset. We evaluate XLM-R\(_{EM}\), null prompts, and our prompt variants. For each result, the mean and standard deviation of 5 runs are reported. \(\times\): macro average across 14 languages. The standard deviations are quite large which indicates that multiple runs are needed and results are seed dependent. In-language prompting provides the most consistent results, with Polish 8-shot as lowest score (31.8 \(F_1\)). Other methods all have results below 15.0 \(F_1\).