Generative Prompt Tuning for Relation Classification

Anonymous ACL submission

Abstract

Prompt tuning is proposed to better tune pre-trained language models by filling the objective gap between the pre-training process and the downstream tasks. Current methods mainly convert the downstream tasks into masked language modeling (MLM) problems, which have proven effective for tasks with simple label sets. However, when applied to relation classification tasks which often exhibit a complex label space, vanilla prompt tuning methods designed for MLM may struggle with handling complex label verbalizations with variable length as in such methods, the locations and number of masked tokens are typically fixed. Inspired by the text infilling task for pre-training generative models that can flexibly predict missing spans, we propose a novel generative prompt tuning method to reformulate relation classification as an infilling problem to eliminate the rigid prompt restrictions, which allows our method to process label verbalizations of varying lengths at multiple predicted positions and thus be able to fully leverage rich semantics of entity and relation labels. In addition, we design entity-guided decoding and discriminative relation scoring to predict relations effectively and efficiently in the inference process. Extensive experiments under low-resource settings and fully supervised settings demonstrate the effectiveness of our approach.

1 Introduction

Relation classification (RC) is a fundamental task in natural language processing (NLP), aiming to detect the relations between the entities contained in a sentence. With the rise of a series of pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020; Raffel et al., 2020), fine-tuning PLMs has become a dominating approach to RC (Joshi et al., 2020; Xue et al., 2021; Zhou and Chen, 2021). However, the significant objective gap between pre-training and fine-tuning may hinder the full potential of pre-trained knowledge for such a downstream task.

To this end, prompt tuning (Brown et al., 2020; Schick and Schütze, 2021a,b; Liu et al., 2021a) has been recently proposed. The core idea is to convert the objective of downstream tasks to be closer to that of the pre-training tasks. Current methods mainly cast a specific task to a masked language modeling (MLM) problem through two components: a template to reformulate input examples into cloze-style phrases (e.g., "<*input example*>. *It was* [MASK]."), and a verbalizer to map labels to candidate words (e.g., *positive*→"*great*" and *negative*→"*terrible*"). By predicting [MASK] ("*great*" or "*terrible*"), we can determine the label of the input example (*positive* or *negative*). Prompt tuning has proven effective especially for low-resource scenarios (Gao et al., 2021; Scao and Rush, 2021) by injecting task-specific guidance. When the label space is simple, downstream tasks can easily adapt to this paradigm (Hambardzumyan et al., 2021; Lester et al., 2021), which predicts one verbalization token at one masked position in the template.

However, when applying prompt tuning to RC with complex label space that conveys rich semantic information, vanilla prompt tuning methods designed for MLM may struggle with handling complex label verbalizations with variable length as in such methods, the locations and number of masked tokens are typically fixed. As presented in Figure 1 (b), different labels involve varying numbers of words as their descriptions. Abridging such labels into verbalizations of fixed lengths requires expert efforts and may lose important label semantic information, which is crucial for RC (Chang et al., 2008; Sainz et al., 2021). The problem becomes more tricky to handle when multiple predicted slots are required in the template, each of which may correspond to varying numbers of words to be predicted. This will hinder injecting essential knowledge such as entity types (Zhou and Chen, 2021) for RC. Fun-

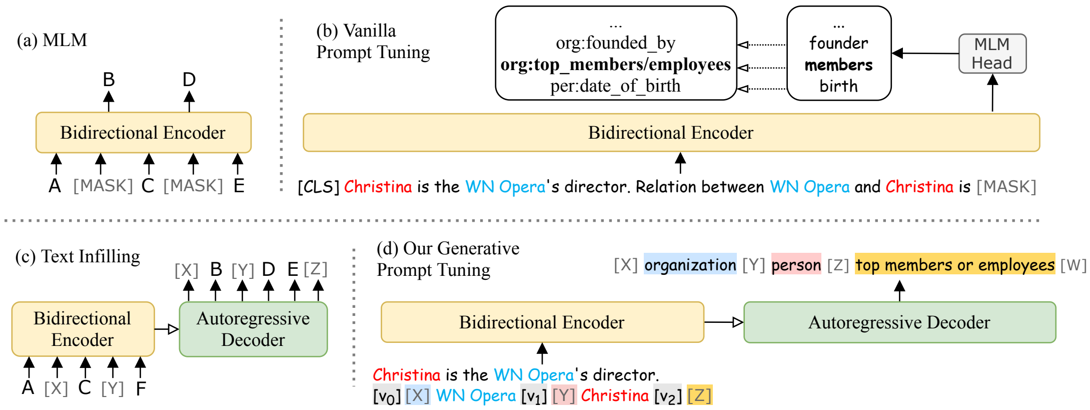

Figure 1: An illustration of (a) MLM pre-training, (b) vanilla prompt tuning intuitively applied to RC, (c) text infilling pre-training, and (d) our proposed generative prompt tuning approach for RC.

damentally, these limitations are because the existing prompt tuning methods imitate MLM, which predicts only one token at one masked position. Therefore, we revisit existing pre-training tasks. As shown in Figure 1 (c), different from MLM, text infilling task (Lewis et al., 2020; Raffel et al., 2020) for pre-training generative models appears to be more compatible with RC. The task replaces consecutive spans of tokens with a single sentinel token and feeds the corrupted sentence into the encoder. The decoder learns to predict not only which but also how many tokens are missing from each span.

Inspired by the text infilling pre-training task, we propose a novel *Gen*erative *P*rompt *T*uning method (GenPT), which eliminates the rigid prompt restrictions and reformulates RC as an infilling task to fully exploit the semantics of entity and relation types. Specifically, we construct an entity-oriented prompt, in which the template converts input sentences to infilling style phrases by leveraging three sentinel tokens, which serve as placeholders for type tokens of head and tail entities and label verbalizations. The target sequence then corresponds to entity type tokens and label verbalizations. In this way, our model can flexibly process label verbalizations of different lengths at multiple predicted positions, so as to fully utilize the semantic information of entity and relation types without the need for manual prompt engineering. Moreover, efficiently deciding the final classes is a practical problem in applying generative models to discriminative tasks. We design a simple yet effective entity-guided decoding and discriminative relation scoring strategy, making the prediction process more robust and efficient.

We conduct extensive experiments on four widely used relation classification datasets under low-resource and fully supervised settings. Compared to a series of strong discriminative and generative baselines, our method achieves better or competitive performance, especially in cases where relations are rarely seen during training, illustrating the effectiveness of our approach.

Our main contributions are as follows[1]:

- We reformulate relation classification as a text infilling task and propose a novel generative prompt tuning method, which eliminates the rigid prompt restrictions and makes full use of semantic information of entity types and relation labels.
- We design entity-guided decoding and discriminative relation scoring strategies to predict relations in the inference process effectively and efficiently.
- Extensive experiments on four popular relation classification datasets demonstrate the effectiveness of our model in both low-resource and fully supervised settings.

2 Background

2.1 MLM and Text Infilling

Masked language modeling (Taylor, 1953) is widely adopted as a pre-training task to obtain a bidirectional pre-trained model (Devlin et al., 2019; Liu et al., 2019; Conneau and Lample, 2019). Generally speaking, a masked language model (MLM) randomly masks out some tokens from the input

[1] We attach our code to the supplement and will release the code at ∗URL∗ once the paper is accepted.

sentences. Each [MASK] corresponds to one word. The objective is to predict the masked word by the rest of the tokens (see Figure 1 (a)).

Different from MLM which only predicts one token for one [MASK], the text infilling task (Raffel et al., 2020; Lewis et al., 2020) for pretraining seq2seq model can flexibly generate spans with different lengths. As shown in Figure 1 (c), the text infilling task samples a number of text spans with different lengths from the original sentence. Then each span is replaced with a single sentinel token. The encoder is fed with the corrupted sequence, and the decoder sequentially produces the consecutive tokens of dropped-out spans delimited by sentinel tokens.

2.2 Prompt-tuning of PLMs

During the standard fine-tuning of classification, the input instance $\boldsymbol{x}$ is converted to a token sequence $\widetilde{\boldsymbol{x}} = \texttt{[CLS]}\boldsymbol{x}\texttt{[SEP]}$. The model predicts an output class by adding a classification head on top of the [CLS] representations. Despite the effectiveness of fine-tuning PLMs, there is a big gap between pre-training tasks and fine-tuning tasks. To this end, prompt-tuning is proposed to convert the downstream task to make it consistent with the pre-training task. Current prompt-tuning approaches mainly cast tasks to cloze-style questions to imitate MLM. Formally, a prompt consists of two key components, template and verbalizer. The template $T(\cdot)$ reformulates the original input $\boldsymbol{x}$ as a cloze-style phrase $T(\boldsymbol{x})$ by adding a set of additional tokens and one [MASK] token. The verbalizer $\phi : \mathcal{R} \to \mathcal{V}$ maps task labels $\mathcal{R}$ to textual tokens $\mathcal{V}$, where $\mathcal{V}$ refers to a set of label words in the vocabulary of a language model $\mathcal{M}$. In this way, a classification task is transformed into a MLM task:

$$P(r \in \mathcal{R}|\boldsymbol{x}) = P(\texttt{[MASK]} = \phi(r)|T(\boldsymbol{x}))$$

Existing prompt-based approaches are effective when the label verbalizations are short with fixed length, but struggle in cases where labels require more complex and elaborate descriptions, as in relation classification. We can see from Figure 1 (b) that different classes own label tokens of different lengths, and it may not always be easy to map them to verbalizations of the same length without losing semantic information.

3 Approach

As presented in Figure 1 (d), this paper considers relation classification as a text infilling style task under a seq2seq framework, which takes the sequence $T(\boldsymbol{x})$ processed by the template as input and outputs a target sequence $\boldsymbol{y}$ to predict relations. This section gives the problem definition formally in Section 3.1 and details our proposed approach. We first introduce how to construct entity-oriented prompts in Section 3.2, and then show the model and training objective in Section 3.3. The inference details including entity-guided decoding and relation scoring are in Section 3.4.

3.1 Problem Definition

Formally, for an instance $\boldsymbol{x} = [x_1, x_2, ..., x_{|\boldsymbol{x}|}]$ with head and tail entity mentions $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$ spanning several tokens in the sequence, as well as entity types $\boldsymbol{t}_h$ and $\boldsymbol{t}_t$, relation classification task is required to predict the relation $r \in \mathcal{R}$ between the entites, where $\mathcal{R}$ is the relation set. $\boldsymbol{r}$ represents the corresponding label verbalization. Take a sentence $\boldsymbol{x}$ = "*Christina is the Washington National Opera's director*" with relation r = "*org:top_members/employees*" as an example, $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$ are "*Christina*" and "*Washington National Opera*", and their entity types are "*organization*" and "*person*" respectively. The relation label verbalization $\boldsymbol{r}$ = "*top members or employees*" are derived from label r, which involves removing attribute words "*org:*", discarding symbols of "_", and replacing "/" with "or".

3.2 Entity-oriented Prompt Construction

We design an entity-oriented continuous template $T(\cdot)$ combining entity mentions and type information, which uses a series of pseudo tokens (Liu et al., 2021c) as prompts rather than discrete token phrases. Specifically, for an input sentence $\boldsymbol{x}$ with two marked entities $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$,

$$\begin{aligned} T(\boldsymbol{x}) = \boldsymbol{x}\, [v_0]...[v_{n_0-1}]\, \texttt{[X]}\, \boldsymbol{e}_h\, [v_{n_0}]...[v_{n_1-1}] \\ \texttt{[Y]}\, \boldsymbol{e}_t\, [v_{n_1}]...[v_{n_2-1}]\, \texttt{[Z]} \end{aligned} \quad (234)$$

where $[v_i] \in \mathbb{R}^d$ refers to the i-th pseudo token in the template. We add three sentinel tokens in the template, where [X] and [Y] in front of entity mentions are expected to denote type information of head and tail entities, and [Z] to represent relation label tokens. The target sequence then consists of head and tail entity types and label verbalizations, delimited by the sentinel tokens used in the input plus a final sentinel token [W].

$$\boldsymbol{y} = \texttt{[X]}\, \boldsymbol{t}_h\, \texttt{[Y]}\, \boldsymbol{t}_t\, \texttt{[Z]}\, \boldsymbol{r}\, \texttt{[W]}$$

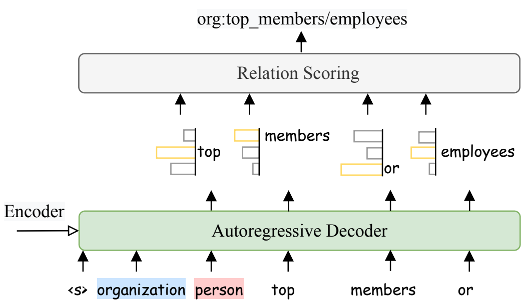

Figure 2: Entity-guided decoding and relation scoring.

where $\boldsymbol{t}_h$ and $\boldsymbol{t}_t$ denote the entity type sequence, $\boldsymbol{r}$ represents the token verbalizations of relation label. For example, we convert the example given in Section 3.1 to :

$$T(\boldsymbol{x}) = \boldsymbol{x}\ [v_0]...[v_{n_0-1}]\ \texttt{[X]}\ \textit{Washington National Opera}\ [v_{n_0}]...[v_{n_1-1}]\ \texttt{[Y]}\ \textit{Christina}[v_{n_1}]...[v_{n_2-1}]\ \texttt{[Z]}$$

The target sequence will be $\boldsymbol{y}$ = "[X], *organization*, [Y], *person*, [Z], *top*, *members*, *or*, *employees*, [W]".

3.3 Model and Training

Given the generative PLM $\mathcal{M}$ and a template $T(\boldsymbol{x})$ as input, we map $T(\boldsymbol{x})$ into embeddings in which the pseudo tokens are mapped to a sequence of continuous vectors,

$$\mathrm{e}(\boldsymbol{x}), h_0, ..., h_{n_0-1}, \mathrm{e}(\texttt{[X]}), \mathrm{e}(\boldsymbol{e}_h), h_{n_0}, ..., h_{n_1-1}, \mathrm{e}(\texttt{[Y]}), \mathrm{e}(\boldsymbol{e}_t), h_{n_1}, ..., h_{n_2-1}, \mathrm{e}(\texttt{[Z]})$$

where $\mathrm{e}(\cdot)$ is the embedding layer of $\mathcal{M}$, $h_i \in \mathbb{R}^d$ are trainable embedding tensors with random initialization, d is the embedding dimension of $\mathcal{M}$, and $0 \leq i < n_2$. We feed the input embeddings to the encoder of the model, and obtain hidden representations of the sentence $\mathbf{h}$:

$$\mathbf{h} = \mathrm{Enc}(T(\boldsymbol{x}))$$

At the j-th step of the decoder, the model attends to previously generated tokens $y_{<j}$ and the encoder output $\mathbf{h}$, and then predicts the probability of the next token:

$$P(y_j|y_{<j}, T(\boldsymbol{x})) = \mathrm{Dec}(y_{<j}, \mathbf{h})$$

We train our model by minimizing the negative log-likelihood of label text $\boldsymbol{y}$ tokens given $T(\boldsymbol{x})$ as input:

$$\mathcal{L} = -\sum_{j=1}^{|\boldsymbol{y}|} \log P(y_j|y_{<j}, T(\boldsymbol{x}))$$

Dataset	#train	#dev	#test	#rel
TACRED	68,124	22,631	15,509	42
TACREV	68,124	22,631	15,509	42
Re-TACRED	58,465	19,584	13,418	40
Wiki80	44,800	5,600	5,600	80

Table 1: Statistics of datasets used.

3.4 Entity-guided Decoding and Scoring

We propose a simple yet effective entity-guided decoding strategy, which exploits entity type information to implicitly influence the choice of possible candidate relations. As shown in Figure 2, at the beginning of decoding, instead of only inputting the *start-of-sequence* token <s> to the decoder, we also append the entity type tokens. With $\hat{\boldsymbol{y}}$ = <s> [X] $\boldsymbol{t}_h$ [Y] $\boldsymbol{t}_t$ [Z] as initial decoder inputs that serves as "preamble", the model iteratively predicts the subsequent tokens:

$$P(y_j|\hat{\boldsymbol{y}}, y_{<j}, T(\boldsymbol{x})) = \mathrm{Dec}(\hat{\boldsymbol{y}}, y_{<j}, \mathbf{h})$$

We collect $\mathbf{P} \in \mathbb{R}^{L \times |\mathcal{V}|}$ through the decoding process, where P_j is word probability at the j-th prediction step, L represents the maximum generation length. The relation is predicted depending on the generated token probability corresponding to relation label verbalizations. Formally, for each relation $r \in \mathcal{R}$ with its label verbalization $\boldsymbol{r}$, the prediction score s_r is calculated as follows:

$$s_r = \frac{1}{|\boldsymbol{r}|} \sum_{j=1}^{|\boldsymbol{r}|} p_{j, \boldsymbol{r}_j}$$

where $p_{j,\boldsymbol{r}_j}$ represents the probability of token $\boldsymbol{r}_j$ at the j-th step of decoding. The sentence is classified into the relation with the highest score.

4 Experiments

4.1 Datasets and Setups

We conduct experiments on four RC datasets, which are TACRED[2] (Zhang et al., 2017), TACREV[3] (Alt et al., 2020), Re-TACRED[4] (Stoica et al., 2021), and Wiki80[5] (Han et al., 2019), as presented in Table 1. TACRED is one of the most widely used RC datasets. TACREV is a dataset

[2] https://nlp.stanford.edu/projects/tacred/
[3] https://github.com/DFKI-NLP/tacrev
[4] https://github.com/gstoica27/Re-TACRED
[5] https://github.com/thunlp/OpenNRE

	Model	#Params	TACRED K=8	TACRED K=16	TACRED K=32	TACREV K=8	TACREV K=16	TACREV K=32	Re-TACRED K=8	Re-TACRED K=16	Re-TACRED K=32	Wiki80 K=8	Wiki80 K=16	Wiki80 K=32
Fine-tuning	SpanBERT* (Joshi et al., 2020)	336M	8.4	17.5	17.9	5.2	5.7	18.6	14.2	29.3	43.9	40.2	70.2	73.6
	LUKE* (Yamada et al., 2020)	483M	9.5	21.5	28.7	9.8	22.0	29.3	14.1	37.5	52.3	53.9	71.6	81.2
	GDPNet† (Xue et al., 2021)	336M	–	–	–	8.3	20.8	28.1	18.8	48.0	54.8	45.7	61.2	72.3
	TANL* (Paolini et al., 2021)	770M	18.1	27.6	32.1	18.6	28.8	32.2	26.7	50.4	59.2	68.5	77.9	82.2
	TYP Marker‡ (Zhou and Chen, 2021)	355M	<u>28.9</u>	32.0	<u>32.4</u>	27.6	31.2	32.0	44.8	54.1	60.0	31.5*	57.0*	77.4*
Prompt Tuning	PTR (Roberta)‡ (Han et al., 2021)	355M	28.1	30.7	32.1	<u>28.7</u>	31.4	32.4	<u>51.5</u>	56.2	<u>62.1</u>	–	–	–
	PTR (BERT)† (Han et al., 2021)	336M	–	–	–	25.3	27.2	33.1	45.8	53.8	55.2	67.6	75.6	78.8
	KnowPrompt† (Chen et al., 2021)	336M	–	–	–	28.6	30.8	34.2	45.8	53.8	55.2	<u>71.8</u>	<u>78.8</u>	81.3
	GenPT (BART)	406M	**29.7** (±0.7)	**33.5** (±0.7)	**35.0** (±0.7)	**29.6** (±0.6)	**32.9** (±0.7)	<u>34.3</u> (±0.4)	50.6 (±2.7)	<u>56.7</u> (±0.8)	<u>62.1</u> (±1.8)	71.4 (±0.4)	78.0 (±0.4)	<u>82.4</u> (±0.6)
	GenPT (T5)	770M	28.2 (±1.3)	<u>32.1</u> (±1.4)	**35.0** (±0.9)	27.9 (±1.8)	<u>31.7</u> (±1.5)	**34.6** (±0.9)	**52.4** (±1.6)	**57.3** (±1.9)	**62.3** (±1.5)	**73.5** (±0.8)	**79.2** (±0.6)	**83.0** (±0.4)

Table 2: Low-resource results on TACRED, TACREV, Re-TACRED, and Wiki80 datasets. We report mean and standard deviation performance of micro F_1 (%) over 5 different splits (see Section 4.1). Results marked with † are reported by Chen et al. (2021), ‡ are reported by (Han et al., 2021), and $\star$ indicates we rerun original code under low-resource settings. **Best** and <u>second best</u> numbers are highlighted in each column.

revised from TACRED, which has the same training data as the original TACRED and extensively relabeled development and test sets. Re-TACRED is another completely re-annotated version of TACRED dataset. Wiki80 is a relation classification dataset derived from FewRel (Han et al., 2018), a large scale few-shot RC dataset. We follow the split used in Chen et al. (2021). The entity type information is obtained from Wikidata (Vrandecic and Krötzsch, 2014).

We evaluate our model under low-resource setting and fully supervised setting. For the low-resource setting, we randomly sample K instances per relation for fine-tuning and validation, with K to be 8, 16, and 32, respectively. Following the work of Gao et al. (2021), we measure the average performance across five different randomly sampled data using a fixed set of seeds for each experiment. We also report the performance under fully supervised setting, where all training and development sets are available.

4.2 Implementation Details

The approach is based on Pytorch (Paszke et al., 2019) and the Transformer library of Huggingface (Wolf et al., 2020). We implement our method on two pretrained transformer language models, T5_{large} (Raffel et al., 2020) and BART_{large} (Lewis et al., 2020). The approach based on T5 is described in detail in Section 3. The BART version is basically the same as the T5 version, except that the sentinel tokens in the template are replaced with `[MASK]` tokens, following the pre-training task format of BART, and the target sequence is composed of entity types and label verbalizations. Most hyper-parameters are chosen following previous works (Han et al., 2021; Zhou and Chen, 2021). The maximum length of input sequence is 512. The maximum generation length L depends on the maximum length of label verbalizations. The length of pseudo tokens in the template $T(\cdot)$ is set to $n \times 3$, where $n_0 = n_1 - n_0 = n_2 - n_1 = n$. n is 3 in our implementation, and detailed discussion is in Section 4.5. During training, our model is optimized with AdamW (Loshchilov and Hutter, 2019) with a learning rate of $3e-5$. We use a batch size of 4 for T5 and 16 for BART, which are chosen for practical consideration in order to fit into GPU memory. The epochs are set to 5 and 10 for fully supervised setting and low-resource setting. The model is trained on 1 NVIDIA Tesla V100 GPU. The training times of TACRED under $K = 16$ and fully supervised settings are 0.36 hours and 10.1 hours, respectively, and testing time is 0.54 hours. We conduct ablation experiments and performance analysis based on the BART version.

4.3 Baselines

We compare our model with some recent efforts. They are 1) SpanBERT (Joshi et al., 2020), a span-based pretraining model, 2) LUKE (Yamada et al., 2020), a pretrained contextualized representations of words and entities based on the Transformer, 3) GDPNet (Xue et al., 2021), constructing a latent multi-view graph to find indicative words from long sequences for RC, 4) TYP Marker (Zhou and Chen, 2021), incorporating entity representations with typed markers, 5) TANL (Paolini et al., 2021), framing structured prediction as a translation task, 6) PTR (Han et al., 2021), a prompt tun-

	Model	TACRED	TACREV	Re-TACRED	Wiki80
Fine-tuning	SpanBERT	70.8	$78.0^{\star}$	$85.3^{¶}$	$88.1^{\star}$
	LUKE	72.7	$80.6^{‡}$	$90.3^{‡}$	$89.2^{\star}$
	GDPNet	70.5	80.2	–	–
	TANL	$72.1^{\star}$	$81.2^{\star}$	$90.8^{\star}$	$89.1^{\star}$
	TYP Marker	**74.6**	**83.2**	**91.1**	**91.3***
Prompt Tuning	PTR (Roberta)	72.4	81.4	90.9	–
	PTR (BERT)	–	$80.2^{†}$	$89.0^{†}$	–
	KnowPrompt	–	80.8	89.8	–
	GenPT (BART)	74.0	82.0	90.3	90.5
	GenPT (T5)	<u>74.1</u>	<u>82.9</u>	<u>91.0</u>	<u>90.6</u>

Table 3: Fully supervised results of micro F_1 (%) on four datasets. $*$ are reported by Alt et al. (2020), ¶ are reported by Stoica et al. (2021), ‡ are reported by Zhou and Chen (2021), † are reported by Chen et al. (2021), $\star$ indicates we rerun original code, and others are from the original papers. **Best** and <u>second best</u> numbers are highlighted in each column.

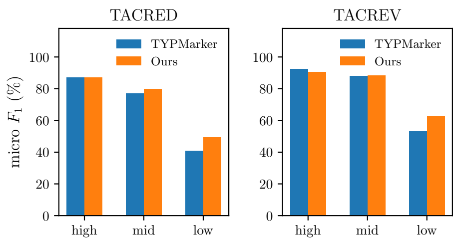

Figure 3: Comparison of F_1 (%) with different frequency relations on TACRED and TACREV.

ing method with rules, which apply logic rules to construct prompts with several sub-prompts, and 7) KnowPrompt (Chen et al., 2021), a knowledge-aware prompt tuning approach that injects knowledge into template design and answer construction. Among these works, LUKE adopts extra data for pre-training, PTR, KnowPrompt, and TYP Marker also utilize entity types in their methods.

4.4 Main Results and Discussion

Results of Low-Resource RC Table 2 presents the results of micro F_1 under low-resource setting. We report mean and standard deviation over 5 different sampled training and development sets. Our model achieves better or comparable performance in comparison to existing approaches. Specifically, our model outperforms the state-of-the-art discriminative fine-tuning model TYP Marker and prompt tuning methods PTR and KnowPrompt, proving that our method can handle extremely few-shot classification tasks better. We compare with generative model TANL which frames relation classification as a translation task. It can be observed that our method outperforms TANL, and the performance gain mainly comes from three aspects: 1) We convert RC to a text infilling task to be consistent with the pre-training task. 2) We fully leverage the entity type information in training and inference to improve RC. 3) Compared to their complex decoding strategy, our relation scoring module is more efficient. See Section 4.6 for more discussion.

Results of Fully Supervised RC As shown in Table 3, we evaluate our model on the fully supervised setting. We can see our method outperforms some strong baselines including SpanBERT, LUKE, and GDPNET, and reaches comparable performance to the recent state-of-the-art model TYP Marker, which incorporates entity types into entity markers and achieves effective representations by concatenating the vectors of entity markers to predict relations. Moreover, we obtain better results on TACRED and TACREV datasets compared to the prompt-tuning model PTR and KnowPrompt. This result illustrates that it is practical to convert the relation classification task to a text infilling task and employ a pre-trained seq2seq model to generate label verbalizations. In this way, we can fully utilize the semantic information of relation labels without the need of manual prompt engineering.

Impact of Training Relation Frequency To further explore the impact of relation frequencies in training data, we split the test set into three subsets according to the class frequency in training. Specifically, we regard the relations with more than 300 training instances to form a high frequency subset (except for "*no_relation*"), those with 50-300 training instances form a middle frequency subset, and the rest form a low frequency subset. The high, middle, and low frequency subsets consist of 11, 25, and 5 relations, with each containing 2,263, 1,024, and 38 instances on TACRED and 2,180, 912, and 31 on TACREV. As shown in Figure 3, we evaluate our model and TYP Marker on the three subsets of the test data. Although the performance of our model is slightly lower than that of the TYP Marker on the high frequency set, we outperform it on the other two subsets, especially on the low frequency set, proving that our model is more effective when the class rarely appears in the training data.

4.5 The Effect of Prompt

The impact of prompt format Extensive experiments with different template formats are conducted to illustrate the effect of prompt construc-

No.	Inputs	Targets	TACRED K=8	TACRED K=16	TACRED K=32
1	$\boldsymbol{x}$ $[v_0]...[v_{n_0-1}]$ `[MASK]` $\boldsymbol{e}_h[v_{n_0}]...[v_{n_1-1}]$ `[MASK]` $\boldsymbol{e}_t[v_{n_1}]...[v_{n_2-1}]$ `[MASK]`	$\boldsymbol{t}_h\ \boldsymbol{t}_t\ \boldsymbol{r}$	**29.7**	**33.5**	**35.0**
2	$\boldsymbol{x}$ $[v_0]...[v_{n_0-1}]$ `[MASK]` $\boldsymbol{e}_h[v_{n_0}]...[v_{n_1-1}]$ `[MASK]` $\boldsymbol{e}_t[v_{n_1}]...[v_{n_2-1}]$ `[MASK]`	$\boldsymbol{t}_h\ \boldsymbol{t}_t\ \boldsymbol{r}$	28.2	31.8	33.6
3	$\boldsymbol{x}$ $[v_0]...[v_{n_0-1}]$ $\boldsymbol{t}_h$ $\boldsymbol{e}_h[v_{n_0}]...[v_{n_1-1}]$ $\boldsymbol{t}_t$ $\boldsymbol{e}_t[v_{n_1}]...[v_{n_2-1}]$ `[MASK]`	$\boldsymbol{r}$	28.1	31.5	33.1
4	$\boldsymbol{x}$ $[v_0]...[v_{n_0-1}]$ $\boldsymbol{e}_h[v_{n_0}]...[v_{n_1-1}]$ $\boldsymbol{e}_t[v_{n_1}]...[v_{n_2-1}]$ `[MASK]`	$\boldsymbol{r}$	27.9	31.2	32.9
5	$\boldsymbol{x}$	$\boldsymbol{t}_h\ \boldsymbol{t}_t\ \boldsymbol{r}$	9.68	12.3	13.3
6	$\boldsymbol{x}$	$\boldsymbol{r}$	9.95	11.6	13.1

Table 4: Ablation study on TACRED showing micro F_1 (%) to illustrate the impact of prompt formats. The shadow in row #1 indicates our entity-guided decoding, and row #2 represents the model without entity-guided decoding.

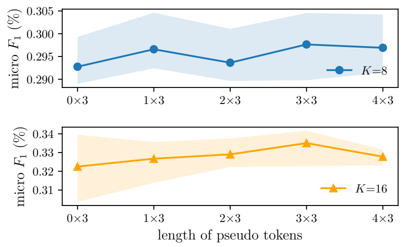

Figure 4: Micro F_1 (%) with different numbers of pseudo tokens on TACRED.

Model	TACRED K=8	TACRED K=16	TACRED K=32
Ours	29.7	33.5	35.0
Handcrafted verbalization	28.1	31.2	32.8

Table 5: Analysis of verbalizations with original label tokens or handcrafted tokens.

Model	TACRED K=8	TACRED K=16
Ours	29.7	33.5
Likelihood-based prediction (LP)	29.6	32.7

Table 6: Micro F_1 (%) and inference time (hours) on the test set with our relation scoring and likelihood-based prediction, respectively.

tion. As shown in row #3 of Table 4, we add entity types to the input sequence instead of predicting them in the targets. Row #4 represents removing the mask tokens corresponding to entity types in the template, and only predicting the relation labels. The F_1 score of the model under three different few-shot settings degraded. Moreover, we compare our model to the vanilla fine-tune pre-trained seq2seq model, where the encoder inputs are original sentences, and targets are relation labels with (row #5) and without (row #6) entity types, respectively. As we can see, our model outperforms the vanilla fine-tuning based approach by a large margin, indicating the effectiveness of our entity-oriented prompt design and tuning.

Discussion of pseudo token length Here we discuss the effect of different pseudo token lengths. The experimental results are shown in figure 4. The microF_1 under the setting of $K = 16$ increases when n increases from 0 to 3, and then decreases slightly. In our experiment, we fix n to 3 to achieve effective performance, that is, there are 9 pseudo tokens in the template.

Analysis of label semantics To verify the benefits coming from the label semantics, we experiment on manually crafted label verbalization with fixed length, following the work of Han et al. (2021). For example, relation "*org:founded_by*" is mapped to [*organization, was, founded, by, person*], and relation "*org:top_members/employees*" is mapped to [*organization, 's, employer, was, person*]. Note all relations are mapped to sequences with exactly 5 tokens. With these label verbalizations that require expert efforts, we apply our model by modifying the template $T(\cdot)$ as

$$T(\boldsymbol{x}) = \boldsymbol{x}\ [v_0]...[v_{n_0-1}]\ \texttt{[MASK]}\ \boldsymbol{e}_h[v_{n_0}]...[v_{n_1-1}]$$
$$\texttt{[MASK]}\ [v_{n_1}]...[v_{n_2-1}]\ \texttt{[MASK]}\ \boldsymbol{e}_t$$

and $\boldsymbol{y}$ to be the mapped sequence. The results are presented in Table 5. Our model obtains higher results, proving our model can make full use of label semantics by learning to predict label verbalizations with varying lengths.

4.6 Analysis of Decoding Strategy

The effect of relation scoring During re-running TANL under K=8 on TACRED, we notice that it takes a long time (86.62 hours) to perform inference on the test set. To illustrate the efficiency of our approach, we compare our relation scoring strategy with likelihood-based prediction (Nogueira dos Santos et al., 2020), which feeds

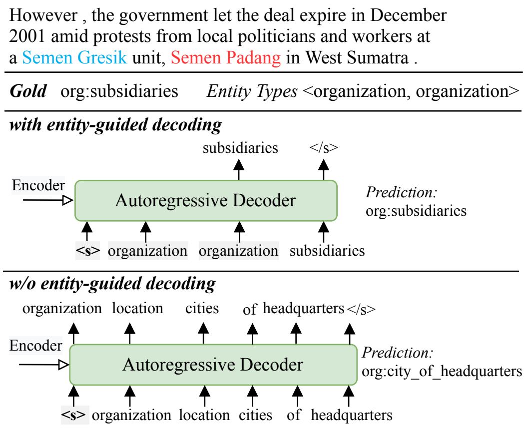

Figure 5: Case study to illustrate the effect of entity-guided decoding.

each candidate sequence into decoder and uses output token likelihoods as corresponding class score. As shown in Table 6, our method achieves promising performance with less inference time.

The effect of entity-guided decoding As shown in row #2 of Table 4, by removing the entity-guided decoding, micro F_1 drops from 35.0 to 33.6 with K=32, proving its effectiveness. We further carry out a detailed case study, shown in Figure 5. A real test instance from TACRED with its entity type information is given. When there is no entity type guidance, the decoder generates the sequence "*organization location cities of headquarters*", and incorrectly classifies the instance as relation "*org:city_of_headquarters*". Our model equipped with entity-guided decoding correctly predicts the relation as "*org:subsidiaries*". This strategy implicitly restricts the generated candidates, gaining performance improvement.

5 Related Work

5.1 Language Model Prompting

Language model prompting has emerged with the introduction of the GPT series (Radford et al., 2018, 2019; Brown et al., 2020). PET (Schick and Schütze, 2021a,b) reformulates input examples as cloze-style phrases and perform gradient-based fine-tuning. ADAPET (Tam et al., 2021) modifies PET's objective to provide denser supervision during fine-tuning. However, these methods require manually designed patterns and label verbalizers. To avoid labor-intensive prompt design, automatic prompt search (Shin et al., 2020; Schick et al., 2020) has been extensively explored. LM-BFF (Gao et al., 2021) adopts T5 to generate prompt candidates and verify their effectiveness through prompt tuning. Continuous prompt learning (Li and Liang, 2021; Qin and Eisner, 2021; Liu et al., 2021c,b) has been further proposed, which directly uses learnable continuous embeddings as prompts rather than discrete token phrases.

5.2 Relation Classification

Fine-tuning PLMs for RC (Joshi et al., 2020; Yamada et al., 2020; Xue et al., 2021; Lyu and Chen, 2021) has achieved promising performance. Zhou and Chen (2021) achieves state-of-the-art results by incorporating entity type information into entity markers. Another interesting line is converting information extraction into generation form, especially when labels have rich semantic information. Zeng et al. (2018) and Nayak and Ng (2020) propose seq2seq models to extract relational facts. Huang et al. (2021) present a generative framework for document-level entity-based extraction tasks. Wang et al. (2021) convert information extraction tasks into a text-to-triple translation framework. A few recent works apply prompt learning on RC. PTR (Han et al., 2021) propose a prompt tuning method with rules by manually designing essential sub-prompts and applying logic rules to compose sub-prompts. KnowPrompt (Chen et al., 2021) design virtual template and answer words with knowledge injected. The main difference between our work and theirs is that we convert RC into an infilling task rather than MLM problem, which can flexibly define templates and label verbalizations by taking advantage of generative models. In addition, our method does not need any manual efforts compared to PTR, which is more practical when adapted to other datasets or similar tasks.

6 Conclusion

This paper presents a novel generative prompt tuning method for RC. Unlike vanilla prompt tuning that converts a specific task into an MLM problem, we reformulate RC as a text infilling task, which can predict label verbalizations with varying lengths at multiple predicted positions and thus better utilize semantic information of entity and relation types. In addition, we design a simple yet effective entity-guided decoding and discriminative scoring strategy, making our generative model more practical. Qualitative and quantitative experiments on four widely used RC benchmarks prove the effectiveness of our approach.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of ACL*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Proceedings of AAAI*.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. *arXiv preprint arXiv:2104.07650*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proceedings of NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of ACL*.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: word-level adversarial reprogramming. In *Proceedings of ACL*.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of EMNLP*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *TACL*, 8:64–77.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of ACL*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. GPT understands, too. *arXiv preprint arXiv:2103.10385*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. In *Proceedings of Findings of ACL*.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of AAAI*.

Cicero Nogueira dos Santos, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Beyond [CLS] through ranking by generation. In *Proceedings of EMNLP*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *Proceedings of ICLR*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of NeurIPS*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of NAACL*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Technical report, OpenAI*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *Technical report, OpenAI*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of EMNLP*.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *Proceedings of NAACL*, pages 2627–2636. Association for Computational Linguistics.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of COLING*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of EACL*.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of NAACL*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of EMNLP*.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the TACRED dataset. In *Proceedings of AAAI*.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of EMNLP*.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP*.

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2021. Gdpnet: Refining latent multi-view graph for relation extraction. In *Proceedings of AAAI*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of EMNLP*.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of ACL*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP*.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *arXiv preprint arXiv:2102.01373*.