Prompt-based Text Entailment for Low-Resource Named Entity Recognition

Dongfang Li[1], **Baotian Hu**[1*], **Qingcai Chen**[1,2*]
[1]Harbin Institute of Technology (Shenzhen), Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
crazyofapple@gmail.com, {hubaotian, qingcai.chen}@hit.edu.cn

Abstract

Pre-trained Language Models (PLMs) have been applied in NLP tasks and achieve promising results. Nevertheless, the fine-tuning procedure needs labeled data of the target domain, making it difficult to learn in low-resource and non-trivial labeled scenarios. To address these challenges, we propose Prompt-based Text Entailment (PTE) for low-resource named entity recognition, which better leverages knowledge in the PLMs. We first reformulate named entity recognition as the text entailment task. The original sentence with entity type-specific prompts is fed into PLMs to get entailment scores for each candidate. The entity type with the top score is then selected as final label. Then, we inject tagging labels into prompts and treat words as basic units instead of n-gram spans to reduce time complexity in generating candidates by n-grams enumeration. Experimental results demonstrate that the proposed method PTE achieves competitive performance on the CoNLL03 dataset, and better than fine-tuned counterparts on the MIT Movie and Few-NERD dataset in low-resource settings.

1 Introduction

Recently, Pre-trained Language Models (PLMs) have achieved promising improvement on several NLP tasks (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020). Nevertheless, fine-tuning language models still needs a moderate number of labeled data for downstream tasks. When difficulties result in limited labeled data available, the trained model shows large variance in downstream performance under full fine-tuning (Mosbach et al., 2021; Le Scao and Rush, 2021). For example, labeling technical and professional terms can be time-consuming and labor-intensive in medical scenarios. Moreover, crowd-sourced annotation is also limited by the reality of existing samples (e.g., when online health assistants are applied to rare diseases).

To address learning challenges in these low-resource scenarios, researchers find that PLMs can learn well by prompt-based learning (Schick and Schütze, 2021a,b; Tam et al., 2021). Prompt-based learning models the probability of text directly; it does not need an extra fully-connected layer usually used by fine-tuning. The main idea is to reformulate NLP tasks as cloze-style question answering for better using the knowledge in PLMs. The model predicts the word probability of masked positions and then derives the final output via mapping relations between words and labels. Previous works have shown the ability of prompt-based learning under low-resource settings (Schick and Schütze, 2021a,b; Schick et al., 2020; Lester et al., 2021). For example, some prompt-based works have explored in classification and generation tasks where it is relatively easy to reformulate into cloze-style tasks (cf. Section 4). Nevertheless, the application to Named Entity Recognition (NER) still poses challenges for current methods. Unlike text classification and text generation, NER is the task of identifying named entities (e.g., *person name*, *location*) in a given sentence, and each unit of the input needs to be predicted. If we directly use Masked Language Modeling (MLM) head to predict each unit label, the lexical and semantic coherence are ignored as there exists latent relationships between the tokens (Lample et al., 2016; Peters et al., 2018; Yan et al., 2019).

In this work, we propose **Prompt-based Text Entailment** (PTE) for low-resource NER. Firstly, we reformulate NER as a *text entailment* task. Textual Entailment (TE) is the task of studying the relation of two sentences, Premise (P) and Hypothesis (H): whether H is true given P (Bowman et al., 2015). Specifically, we treat the original sentence as premise and entity type-specific prompt as a hypothesis. Given an entity type, the P and H are

* Corresponding authors

Proceedings of the 29th International Conference on Computational Linguistics, pages 1896–1903
October 12–17, 2022.

fed into PLMs to get entailment scores for each candidate. Then, the entailment score is the probability of a specific token at the mask position of the prompt. After that, the entity type with the top entailment score is selected as the final label. During inference, we enumerate all possible text spans or words in the input sentence as named entity candidates (Cui et al., 2021). The reformulation provides a unified entailment framework for NER tasks where annotations are insufficient, as the model shares the same inference pattern across different domains. As such, we can also leverage generic text entailment datasets such SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) to pre-train models, which transfer knowledge from the general domain and get better performance in new domains. Our method can be a step forward towards the development of a solution for the low-resource NER because any new domain does not typically have extensive annotated data in the real world, whereas it is feasible to obtain a couple of examples (e.g., online assistant). Moreover, considering the existence of noisy annotations, *TE only needs to specify the labels of certain entities for training rather than the complete annotations of the entire sequence*. Experimental results demonstrate that the proposed method `PTE` achieves competitive F1 score on the CoNLL03 dataset (Tjong Kim Sang, 2002), and better than fine-tuned counterparts by a large margin on the MIT Movie (Liu et al., 2013) and Few-NERD datasets (Ding et al., 2021b) in low-resource settings.

2 Method

2.1 Low-Resource Named Entity Recognition

Given a sentence $\mathbf{X} = (x_1, x_2, \dots x_N)$ which contains N words, the task is to produce $\mathbf{Y} = (y_1, y_2, \dots y_N)$ which is the sequence of entity tags. The tag $y_i \in \mathcal{Y}$ (e.g., B-LOC, I-PER, O) denotes the type of entity for each word x_i, where $\mathcal{Y}$ is a pre-defined set of tags. We are given a low-resource NER dataset $\mathcal{D}_{\text{train}}$, where the labeled examples to each NER type (e.g., < 50) are substantially less than that in the rich-resource NER dataset. Our goal is to train an accurate NER model under this low-resource setting.

Previous methods usually treat NER as a sequence labeling task in which a neural encoder such as LSTM and BERT is used for representing the input sequence, and a softmax or a CRF layer is equipped as the output layer to get the tag sequence. Formally, as the standard fine-tuning, NER model $\mathcal{M}$ parameterized by θ is trained to minimize the cross-entropy loss over token representations $\mathbf{H} = [h_1, h_2, \dots h_N]$ that are generated from the neural encoder as follows:

$$\mathcal{L} = -\sum_{i=1}^{N} \log f_{x_i, y_i}(\mathbf{h}; \theta), \tag{1}$$

where f is the model's predicted conditional probability for golden label.

2.2 Prompt-based Text Entailment

Towards the low-resource NER task, a common way is to pre-train the neural encoder and output layer parameters with the rich-resource NER dataset. Another feasible way is to focus on the matching function learned by prototype-based network (Snell et al., 2017) or nearest neighbor classification (Yang and Katiyar, 2020). After that, a well-trained matching function can work well in the target tasks. However, since the entity category is different, the parameter for the low-resource domain cannot be transferred directly from the source domain. Moreover, the metric-based meta-learning methods assume that training and test tasks are in the same distribution but this assumption may not always be satisfied in practice (Yin, 2020).

In this work, we reformulate named entity recognition as the text entailment task. As the NER task is not a standard entailment problem, we convert NER examples into labeled entailment instances. The input includes the original sentence as premise and entity type-specific prompt as a hypothesis (i.e., template). The output is produced by an entailment classifier, predicting a label for each instance. The entailment score is the probability of a specific token at the mask position of the prompt. Then, the entity type with the top entailment score is selected as the final label. For example, given a sentence "*Seoul is the capital of South Korea.*" and a candidate "*Seoul*", we define "*Seoul is an <entity_type> entity. [MASK]*" as prompt for each entity type. Suppose the entailment score of token "*yes*" at [MASK] for <location> type is the highest of all entity types, we finally choose "*location*" as the predicted label. For training, we sample three types of negative examples (see Appendix A): false positive (i.e., replace the correct label with others), null label (i.e., replace the correct label with null), and non-entity replacement (i.e., replace golden entity with non-entity span). For example, "*Seoul is not a*

named entity. [MASK]" is one prompt of "false positive" example (i.e., the [MASK] label is *no*, and it exists entities). During training and inference, we can enumerate all possible text spans in the input sentence as named entity candidates (Cui et al., 2021). To further reduce time complexity in generating candidates by n-grams enumeration, we inject tagging labels (e.g., I-location means the tag is inside a entity) into prompts and treat words as basic units instead of text spans during training and inference. In other words, we consider prompts "*<candidate_entity_word> is the part of a <entity_type> entity. [MASK]*" (e.g., "*Seoul is the part of a location entity. [MASK]"*). As PTE treats words as basic units for decoding, it optimizes time complexity at inference to O(L), which is in line with previous NER methods. It optimizes quadratic costs at inference to linear. We also apply the Viterbi algorithm at inference, where transitions are computed on the training set (Hou et al., 2020). The computational complexity of n-grams enumeration is O(L^2), increasing quadratically with sequence length L. Overall, our method provides a unified entailment framework as the model shares the same inference pattern across different domains.

2.3 Pattern Exploiting Training Framework

The basic framework of PTE is from ADAPET (Tam et al., 2021) which is a variant of PET (Schick and Schütze, 2021a,b). Compared with PET, ADAPET uses more supervision by decoupling the losses for the label tokens and a label-conditioned MLM objective over the total original input (Tam et al., 2021). We introduce it by describing how to convert one example into a cloze-style question. The query-form in ADAPET is defined by a Pattern-Verbalizer Pair (PVP). Each PVP consists of one pattern which describes how to convert the inputs into a cloze-style question with masked out tokens, and one verbalizer which describes the way to convert the classes into the output space of tokens. The PVP can be manually generated (Sun et al., 2019; Petroni et al., 2019) or obtained by using an automatic search algorithm (Schick et al., 2020; Gao et al., 2021). After that, ADAPET obtains logits from the model $G_m(x)$. Given the space of output tokens $\mathcal{Y}$, ADAPET computes a softmax over $y \in \mathcal{Y}$, using the logits from $G_m(x)$. The final loss is shown as follows:

$$q(y|x) = \frac{\exp([\![G_m(x)]\!]_y)}{\sum_{y' \in \mathcal{Y}} \exp([\![G_m(x)]\!]_{y'})}, \quad (2)$$

$$\mathcal{L} = \texttt{Cross\_entropy}(q(y^*|x), y^*). \quad (3)$$

2.4 Cross Task and Domain Transfer

To address the challenge when few labeled examples are available, we further train the sentence encoder on the TE datasets (e.g., MNLI) and apply it to the NER task. Then, our method can perform more knowledge transfer between the rich-resource NER dataset and the low-resource NER dataset. Since there is no domain-related fully connected layer for fine-tuning, all parameters can be transferred in different domains even if the entity category does not match. Specially, we apply the text entailment method to the low-resource domain after firstly pre-training the NER model in the rich-resource domain. This process is simple but can effectively transfer label knowledge. As the output of our method is model-agnostic words (not tag index), the tag vocabulary with rich-resource and low-resource is a shared pre-trained language model vocabulary set. It allows our method to use the correlation of tags to enhance the effect of cross-domain transfer learning.

3 Experiments

We compare our methods with several baselines on both rich-resource settings and low-resource settings. We use the CoNLL2003 (Tjong Kim Sang, 2002) as the rich-resource dataset, and MIT Movie (Liu et al., 2013), Few-NERD (Ding et al., 2021b) as the cross-domain low-resource datasets. And we conduct experiments on the CoNLL03 dataset in both full and low-resource settings. The dataset statistics and experimental settings are included in Appendix B and C. The standard precision, recall, and F1 score are used for model evaluation.

3.1 Rich-Resource NER Results

We first use the whole training set of the CoNLL03 to train the model and evaluate its performance on the test set. Table 1 shows the performance of the comparison method and our model on the test set. We can find that although the potential applications of PTE is low-resource named entity recognition, it can also achieve competitive performance in rich-resource domain data sets. Compared with BERT fine-tuning reported in the previous work, the PTE

Method	Precision	Recall	F1
Wiseman and Stratos (2019)	-	-	89.94
Yang et al. (2018)	-	-	90.77
Ma and Hovy (2016)	-	-	91.21
BERT (Cui et al., 2021)	91.93	91.54	91.73
Yamada et al. (2020)	-	-	**94.30**
Template BART (Cui et al., 2021)	90.51	93.34	91.90
PTE (discrete)	91.27	91.56	91.41
PTE (soft)	92.01	92.45	**92.23**

Table 1: Model performance on the CoNLL03 test set.

Method	PER	ORG	LOC	MISC	Overall
BERT	75.71	77.59	60.72	60.39	69.62
Template BART	84.49	72.61	71.98	73.37	75.59
PTE (BERT)	85.34	72.89	73.01	74.32	**76.40**

Table 2: Cross entity type results on the CoNLL03. LOC and MISC are low-resource entity types, where PER and ORG are rich-resource entity types.

model using discrete manual design reduces the F1 by 0.32, while the PTE model using the soft prompt method design mode (Liu et al., 2021a,b) increases the F1 by 0.5. It shows that our method effectively recognizes named entities, and soft prompts can improve performance compared with manually designed prompts. More experimental results about TE patterns (§2.3) are in the Appendix D.

3.2 Cross Entity Type NER Results

Following Cui et al. (2021), we sample the number of examples corresponding to different types of entities on the CoNLL03 data training set as new training set while keep test set unchanged. Among them, "PER" and "ORG" are rich-resource entity types, and "LOC" and "MISC" are low-resource entity types. The experimental results are shown in Table 2. The results show that our method achieves better results than baselines on the low-resource entity types, thus improving overall performance. On the other hand, our method is better than fine-tuning in both cases.

3.3 Domain Transfer for Low-Resource NER

We do not use N-way K-shot setting (Yang and Katiyar, 2020; Ding et al., 2021b) which samples N categories and K examples for training in each episode because a sentence in the NER task may contain multiple entities from different types. Thus, we randomly sample training data from the MIT Movie and Few-NERD datasets to simulate low-resource scenarios and use CoNLL03 as the rich-resource dataset. As such, we have only K examples for each type of training. We choose $K \in \{10, 20, 50, 100, 200, 500\}$ for experiments to evaluate the ability of the model on training

MIT Movie (12)						
Method	**K=10**	**K=20**	**K=50**	**K=100**	**K=200**	**K=500**
Wiseman and Stratos (2019)	3.1	4.5	4.1	5.3	5.4	8.6
Ziyadi et al. (2020)	40.1	39.5	40.2	40.0	40.0	39.5
Sequence Labeling BERT	28.3	45.2	50.0	52.4	60.7	76.8
Yamada et al. (2020)	35.6	49.2	61.8	72.4	78.7	82.8
Template BART (Cui et al., 2021)	42.4	54.2	59.6	65.3	69.6	80.3
PTE (discrete)	46.9†	59.2†	66.9†	74.9†	79.9†	83.6
PTE (soft)	**47.8**†	**60.8**†	**68.1**†	**76.5**†	**83.6**†	**86.4**†
Few-NERD (8)						
Method	**K=10**	**K=20**	**K=50**	**K=100**	**K=200**	**K=500**
Wiseman and Stratos (2019)	5.2	4.1	4.7	7.8	12.3	10.1
Ziyadi et al. (2020)	35.4	48.3	51.2	51.8	53.6	55.7
Sequence Labeling BERT	50.6	59.3	61.3	61.4	62.5	66.4
Yamada et al. (2020)	51.7	60.1	62.3	61.0	62.5	66.8
PTE (discrete)	51.8	59.7	60.5	61.3	61.8	63.4
PTE (soft)	**54.2**	**61.4**	**62.3**	**62.5**	**63.6**	**67.4**

Table 3: F1 comparison of two low-resource NER datasets. We set 6 sample size K for different low-resource settings. † means a significant difference compared to Template BART ($p < .05$).

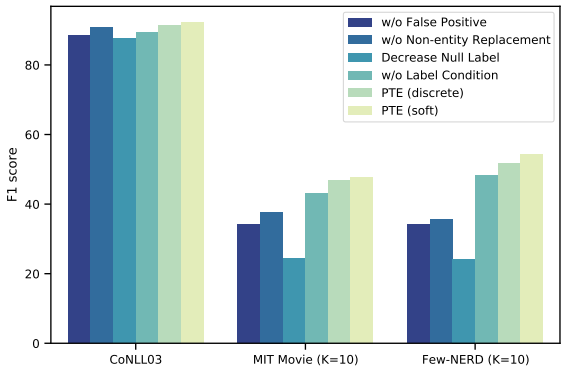

Figure 1: F1 scores with different experimental settings and model variants.

data of different sizes. The experimental results are in Table 3. The results show that when the K value is relatively small, our PTE method can be better than the fine-tuning method, and this trend decreases with the increase of K. In addition, the soft mode is also better than the discrete mode in the case of a small number of samples. Overall, our method achieves the best results on both data sets in the low-resource scenario.

3.4 Ablation Study

We conduct ablation experiments and the results are shown in Figure 1. The results show that (1) the selection of negative examples has a great impact on the performance of the model, especially the negative examples of the null label type. However, in rich-resource scenario, the gap between full setting and decreased setting is not as much as the low-resource scenario; (2) the low-resource scenario is a challenge to the model, and the results of some variants are not inconsistent where prompt-based learning may not be as good as fine-tuning; (3) label conditioning and soft mode have a consistent effect on the model. These findings highlight that it still has room left to use prompt

for effectively transferring knowledge in the case of low-resource scenario.

4 Related Work

Previous works have shown the ability of prompt-based learning under low-resource settings(Schick and Schütze, 2021a,b; Schick et al., 2020; Lester et al., 2021). Schick and Schütze (2021a) address low-resource text classification by manually designing templates as prompt-based learning in a iterative training manner. Gao et al. (2021) improve low-resource performance with well-designed templates with demonstrations. Liu et al. (2021b) apply continuous prompts for low-resource learning. Recently, some works (Ding et al., 2021b; Tong et al., 2021; Ma et al., 2021a; Chen et al., 2021) also focus on low-resource NER. In contrast, we propose to use prompt-tuning to treat NER as the TE task. Unlike traditional NER methods, we use prompt-based learning without an additional linear layer for fine-tuning. By defining different prompts, the model is able to perform well in low-resource settings, which adapts to new domains with few labeled data. In contrast to recent work which also adopts prompt-based fine-tuning for NER (Ma et al., 2021b), we show that the effectiveness of the text entailment reformulation for named entity recognition using PLMs.

5 Conclusion

In this paper, we apply prompt-based learning to low-resource named entity recognition. For token classification of NER, we reformulate it into a text entailment task. Our method transfers knowledge in different NLP tasks and domains, and performs better in low-resource scenarios. Future work includes how to apply `PTE` to other NLP tasks.

Acknowledgements

We thank the anonymous reviewers for their insightful comments and suggestions. This work is jointly supported by grants: Natural Science Foundation of China (No. 62006061,61872113,U1813215), Stable Support Program for Higher Education Institutions of Shenzhen (No. GXWD20201230155427003-20200824155011001) and Strategic Emerging Industry Development Special Funds of Shenzhen(No. JCYJ20200109113441941 and No. XMHT20190108009).

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proc. of EMNLP*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Xiang Chen, Ningyu Zhang, Lei Li, Xin Xie, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Lightner: A lightweight generative framework with prompt-guided attention for low-resource NER. *ArXiv preprint*, abs/2109.00720.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021a. Prompt-learning for fine-grained entity typing. *ArXiv preprint*, abs/2108.10604.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021b. Few-NERD: A few-shot named entity recognition dataset. In *Proc. of ACL*, pages 3198–3213, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proc. of ACL*, pages 3816–3830, Online. Association for Computational Linguistics.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proc. of ACL*, pages 1381–1393, Online. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proc. of NAACL-HLT*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proc. of ICLR*.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

J. Liu, P. Pasupat, S. Cyphers, and J. Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390.

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

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *ArXiv preprint*, abs/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*, abs/1907.11692.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021a. Template-free prompt tuning for few-shot NER. *ArXiv preprint*, abs/2109.13532.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021b. Template-free prompt tuning for few-shot NER. *CoRR*, abs/2109.13532.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proc. of ACL*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *Proc. of ICLR*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL-HLT*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proc. of EMNLP*, pages 2463–2473, Hong Kong, China. 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 the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proc. of NAACL-HLT*, pages 380–385, Minneapolis, Minnesota. Association for Computational Linguistics.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4980–4991. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Meihan Tong, Shuai Wang, Bin Xu, Yixin Cao, Minghui Liu, Lei Hou, and Juanzi Li. 2021. Learning from miscellaneous other-class words for few-shot named entity recognition. In *Proc. of ACL*, pages 6236–6247, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of NAACL-HLT*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proc. of ACL*, pages 5363–5369, Florence, Italy. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proc. of EMNLP*, pages 6442–6454, Online. Association for Computational Linguistics.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: adapting transformer encoder for named entity recognition. *ArXiv preprint*, abs/1911.04474.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proc. of EMNLP*, pages 6365–6375, Online. Association for Computational Linguistics.

Wenpeng Yin. 2020. Meta-learning for few-shot natural language processing: A survey. *ArXiv preprint*, abs/2007.09604.

Morteza Ziyadi, Yuting Sun, Abhishek Goswami, Jade Huang, and Weizhu Chen. 2020. Example-based named entity recognition. *ArXiv preprint*, abs/2008.10570.

Datasets	Domain	# Type	# Tokens	# Train	# Dev	# Test
CoNLL03	Reuters news stories	4	21.0k	14041	3250	3453
MIT Movie	Movie reviews	12	6.0k	7820	1955	2443
Few-NERD	Wikipedia	8	4601.2k	131767	18824	37648

Table 4: Statistics of our datasets. We count the number of sentences in the training/development/test set, the number of tokens and the number of tags in datasets.

Type	Templates
Positive (Y)	<candidate> is the part of a <entity_type> entity.
False positive (N)	<candidate> is the part of a <another_entity_type> entity.
Non-entity (N)	<others> is the part of a <entity_type> entity.
Null label (Y/N)	<others> is not a name entity. <candidate> is not a name entity.

Table 5: The discrete manually-crafted templates.

Number	Patterns
Pattern#1	[HYPOTHESIS] ? </s></s> [MASK], [PREMISE] </s>
Pattern#2	" [HYPOTHESIS] " ? </s></s> [MASK], " [PREMISE] " </s>
Pattern#3	[HYPOTHESIS] ? </s></s> [MASK]. [PREMISE] </s>
Pattern#4	" [HYPOTHESIS] " ? </s></s> [MASK]. " [PREMISE] " </s>

Table 6: We list the patterns used by our method where <s> and </s> are start token and separated token.

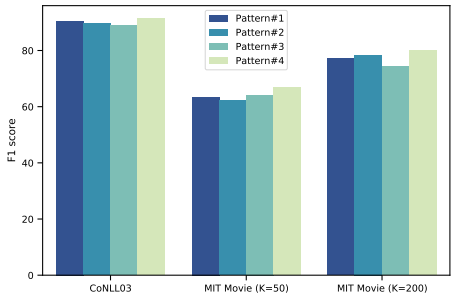

Figure 2: The performance of different modes after up to 7000 training batches. The patterns we use are from RTE task of Tam et al. (2021).

A Templates

We use the naive random sampling method and the positive-negative ratio is 1:1.5 in the low-resource scenario after sampling. As shown in Table 5, we list our templates for each example type used by PTE (discrete). Our soft prompt is to add different special tokens before the [MASK] to form a template and tune the embeddings of these tokens directly following Ding et al. (2021a) used by PTE (soft). We leave it for future work to examine whether the NER performance further improves with a more well-designed soft prompt.

B Dataset Statistics

We use the following datasets where data statistics are displayed in Table 4: (1) The CoNLL03 dataset (Tjong Kim Sang, 2002) is from the English Reuters News and consists of 4 entity types. We use the previous split in Cui et al. (2021) for our experiments. The entity types are *person*, *location*, *organization*, and *miscellaneous entities*. The sampled training dataset in §3.2 includes 1500 organization entities, 1500 person entities, 150 location entities and 150 miscellaneous entities (2) The MIT Movie dataset (Liu et al., 2013) is from queries related to movie information. The entity types are *actor*, *character*, *director*, *genre*, *plot*, *year*, *soundtrack*, *opinion*, *award*, *origin*, *quote*, and *relationship*. (3) The Few-NERD dataset (Ding et al., 2021b) is a low-resource NER dataset with a hierarchy of 8 coarse-grained and 66 fine-grained entity types. We use the coarse-grained entity in our experiments. The entity types are *location*, *event*, *building*, *art*, *product*, *person*, *organization*, and *miscellaneous entities*.

C Experimental Settings

We use the pre-trained models and codes provided by ADAPET and follow their default hyperparameter settings unless noted otherwise. The pre-trained language model of our method is BERT that is pre-trained in the MNLI datasets. We use AdamW optimizer and grid search batch size of $\{8,16,32\}$ for model training. We use grid search for learning rate from $[1e\text{-}5, 2e\text{-}5, 3e\text{-}5, 4e\text{-}5, 5e\text{-}5]$. And we grid search the optimal weight decay weight from $[0.1, 0.01, 0.005, 0.001]$. The maximum sequence length, the dropout rate, the gradient accumulation steps, the maximum training steps and the warm-up ratio are set to 256, 0.1, 16, 7000, 0.06 respectively. Early stopping is also applied based on model performance on the development set. Our models are trained with NVIDIA Tesla V100s. The verbalizer words are ["yes", "no"] and ["true", "false"]. The τ of transition probability in decoding is selected by searching with 0.05 step from 0 to 1. For sequence labeling BERT fine-tuning, we train BERT with a softmax classifier following Devlin et al. (2019), updating parameters using Adam with an initial learning rate of 1e-5, and a batch size of 32.

D Pattern Engineering

After designing templates of entity-specific hypothesis, we follow Tam et al. (2021) to define the TE patterns in Table 6 and report results across all patterns for all datasets in Figure 2. We find that the subtle difference of the prompts impacts performance, while Pattern#4 outperforms others across datasets and settings.