Prompt-Based Metric Learning for Few-Shot NER

Yanru Chen[1], **Yanan Zheng**[1], **Zhilin Yang**[123*]
[1]Tsinghua University, [2]Shanghai Artificial Intelligence Laboratory, [3]Shanghai Qizhi Institute
{achen.cyanr.qaq, zyanan93}@gmail.com
zhiliny@tsinghua.edu.cn

Abstract

Few-shot named entity recognition (NER) targets generalizing to unseen labels and/or domains with few labeled examples. Existing metric learning methods compute token-level similarities between query and support sets, but are not able to fully incorporate label semantics into modeling. To address this issue, we propose a simple method to largely improve metric learning for NER: 1) multiple prompt schemas are designed to enhance label semantics; 2) we propose a novel architecture to effectively combine multiple prompt-based representations. Empirically, our method achieves new state-of-the-art (SOTA) results under 16 of the 18 considered settings, substantially outperforming the previous SOTA by an average of 9.12% and a maximum of 34.51% in relative gains of micro F1. Our code is available at https://github.com/AChen-qaq/ProML.

1 Introduction

Named entity recognition (NER) is a key natural language understanding task that extracts and classifies named entities mentioned in unstructured texts into predefined categories. Few-shot NER targets generalizing to unseen categories by learning from few labeled examples.

Recent advances for few-shot NER use metric learning methods which compute the token-level similarities between the query and the given support cases. Snell et al. (2017) proposed to use prototypical networks that learn prototypical representations for target classes. Later, this method was introduced to few-shot NER tasks (Fritzler et al., 2019; Hou et al., 2020). Yang and Katiyar (2020) proposed StructShot, which uses a pretrained language model as a feature extractor and performs viterbi decoding at inference. Das et al. (2022) proposed CONTaiNER based on contrastive learning. This approach optimizes an objective that characterizes the distance of Gaussian distributed embeddings under the metric learning framework.

Despite the recent efforts, there remain a few critical challenges for few-shot NER. First of all, as mentioned above, metric learning computes token-level similarities between the query and support sets. However, the architectures used for computing similarities in previous work are agnostic to the labels in the support set. This prevents the model from fully leveraging the label semantics of the support set to make correct predictions. Second, while prompts have been demonstrated to be able to reduce overfitting in few-shot learning (Schick and Schütze, 2020), due to a more complex sequence labeling nature of NER, the optimal design of prompts remains unclear for few-shot NER.

In light of the above challenges, we explore a better architecture that allows using prompts to fully leverage the label semantics. We propose a simple method of Prompt-based Metric Learning (ProML) for few-shot NER, as shown in Figure 1. Specifically, we introduce mask-reducible prompts, which is a special class of prompts that can be easily reverted to the original input by using a mask. By performing a masked weighted average over the representations obtained from multiple prompts, our method accepts multiple choices of prompts as long as they are mask-reducible. These prompts improve label efficiency by inserting semantic annotations into the text inputs. As instantiations of this framework, we design an option prefix prompt to provide the model with the candidate label options, and a label-aware prompt to associate each entity with its entity type in the input. As shown in Figure 2, a single prompt provides useful information but has some shortcoming. However, with a weighted average, multiple prompts are combined, which fully leverages label information.

In our experiments, we find that using multiple prompts with the masked weighted average is effective for few-shot NER. Empirically, our method

[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 7199–7212
July 9-14, 2023 ©2023 Association for Computational Linguistics

achieves new state-of-the-art (SOTA) results under 16 of the 18 considered settings, substantially outperforming the previous SOTA by an average of 9.12% and a maximum of 34.51% in relative gains of micro F1.

2 Related Work

Few-Shot NER. Few-shot NER targets generalizing to unseen categories by learning from few labeled examples. Noisy supervised methods (Huang et al., 2020) perform supervised pretraining over large-scale noisy web data such as WiNER (Ghaddar and Langlais, 2017). Self training methods (Wang et al., 2021) perform semi-supervised training over a large amount of unlabelled data. Alternative to these data-enhancement approaches, metric learning based methods have been widely used for few-shot NER (Fritzler et al., 2019; Yang and Katiyar, 2020; Das et al., 2022). Recently, prompt-based methods (Ma et al., 2021; Cui et al., 2021; Lee et al., 2022) are proposed for few-shot NER as well. To introduce more fine-grained entity types in few-shot NER, a large-scale human-annotated dataset Few-NERD (Ding et al., 2021) was proposed. Ma et al. (2022b); Wang et al. (2022) formulate NER task as a span matching problem and decompose it to several procedures. Ma et al. (2022b) decomposed the NER task into span detection and entity typing, and they separately train two models and finetune them on the test support set, achieving SOTA results on Few-NERD (Ding et al., 2021). Different from the above related works, our approach is a general framework of using prompts for token-level metric learning problems.

Meta Learning. The idea of meta learning was first introduced in few-shot classification tasks for computer vision, attempting to learn from a few examples of unseen classes. Since then metric-based methods have been proposed, such as matching networks (Vinyals et al., 2016) and Prototypical networks (Snell et al., 2017), which basically compute similarities according to the given support set, learn prototypical representations for target classes, respectively. It has been shown that these methods also enable few-shot learning for NLP tasks such as text classification (Bao et al., 2019; Geng et al., 2019), relation classification (Han et al., 2018), named entity recognition (Fritzler et al., 2019; Yang and Katiyar, 2020; Das et al., 2022), and machine translation (Gu et al., 2018). Our approach also falls into the category of metric-based meta learning and outperforms previous work on NER with an improved architecture.

Label Semantics for NER. There have been some approaches that make use of label semantics (Ma et al., 2022a; Hou et al., 2020). Hou et al. (2020) propose a CRF framework with label-enhanced representations based on the architecture of Yoon et al. (2019). However, they mainly focus on slot tagging tasks while their performance on NER tasks is poor. Ma et al. (2022a) introduce label semantics by aligning token representations with label representations. Both of them only use label semantics for learning better label representations. In contrast, our approach incorporates label semantics into the inputs so that the model is able to jointly model the label information and the original text samples. This makes the similarity scores dependent on the support set labels and is particularly crucial for metric learning. Our experiments also verify the advantages of our approach compared to previous work using labels semantics.

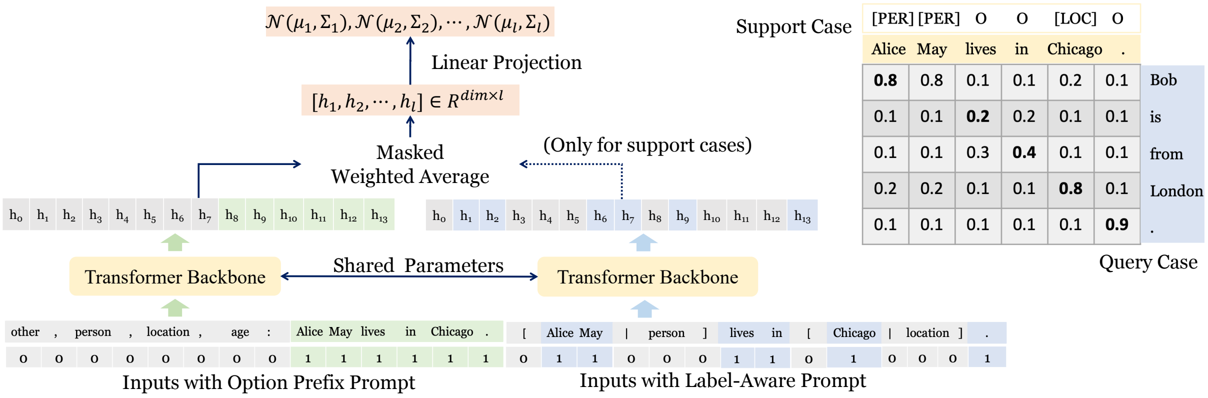
Prompt-Based Approaches for NER. With the emergence of prompt-based methods in NLP research, very recently, some prompt-based approaches for few-shot NER have been proposed (Cui et al., 2021; Lee et al., 2022; Ma et al., 2021). However, they use prompts to help with the label predictions based on classification heads instead of metric learning. Moreover, some of these methods require searching for templates (Cui et al., 2021), good examples (Lee et al., 2022), or label-aware pivot words (Ma et al., 2021), which makes the results highly dependent on the search quality. Different from these methods, our approach does not rely on a search process. More importantly, another key difference is that we employ prompting in the setting of metric learning.

3 Task Definition

3.1 Few-shot NER

Named entity recognition (NER) is a sequence labeling task[1]. Formally, for a sentence $\mathbf{x}$ consisting of n tokens $\mathbf{x} = [x_1, x_2, \cdots, x_n]$, there is a corresponding ground-truth label sequence $\mathbf{y} = [y_1, y_2, \cdots, y_n]$ where each y_i is an encoding of some label indicating the entity type for token x_i. Then a collection of these $(\mathbf{x}, \mathbf{y})$ pairs form a

[1] There also exist other formulations such as span prediction or question answering.


$\mathcal{N}(\mu_1,\Sigma_1), \mathcal{N}(\mu_2,\Sigma_2), \cdots, \mathcal{N}(\mu_l,\Sigma_l)$
Linear Projection
$[h_1, h_2, \cdots, h_l] \in R^{dim \times l}$
Masked
Weighted Average
(Only for support cases)
Support Case
[PER] [PER] O O [LOC] O
Alice May lives in Chicago .
Bob
is
from
London
.
Query Case
Transformer Backbone
Shared Parameters
Transformer Backbone
other , person , location , age : Alice May lives in Chicago .
[Alice May | person] lives in [Chicago | location] .
Inputs with Option Prefix Prompt
Inputs with Label-Aware Prompt

Alice	May	lives	in	Chicago	.	
0.8	0.8	0.1	0.1	0.2	0.1	Bob
0.1	0.1	0.2	0.2	0.1	0.1	is
0.1	0.1	0.3	0.4	0.1	0.1	from
0.2	0.2	0.1	0.1	0.8	0.1	London
0.1	0.1	0.1	0.1	0.1	0.9	.

Figure 1: An overview of the architecture of our proposed ProML . The prompts associated with the input sequence are passed through a transformer backbone to obtain intermediate representations. A masked weighted average is then applied to produce token-level representations. Following Das et al. (2022), Gaussian embeddings for each token are produced using linear projections. The similarity scores between query tokens and support tokens are then computed according to the distance metric.

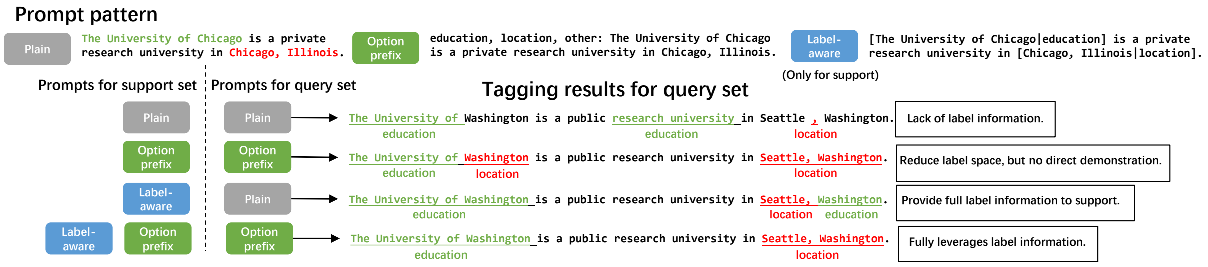

Prompt pattern
Plain
The University of Chicago is a private research university in Chicago, Illinois.
Option prefix
education, location, other: The University of Chicago is a private research university in Chicago, Illinois.
Label-aware
[The University of Chicago|education] is a private research university in [Chicago, Illinois|location].
(Only for support)
Prompts for support set
Prompts for query set
Tagging results for query set
Plain
Plain
The University of Washington is a public research university in Seattle , Washington.
education
education
location
Lack of label information.
Option prefix
Option prefix
The University of Washington is a public research university in Seattle, Washington.
education
location
location
Reduce label space, but no direct demonstration.
Label-aware
Plain
The University of Washington is a public research university in Seattle, Washington.
education
location
education
Provide full label information to support.
Label-aware
Option prefix
Option prefix
The University of Washington is a public research university in Seattle, Washington.
education
location
Fully leverages label information.

Figure 2: A manually constructed example to illustrate different prompts. Prompted inputs for the support set are listed at the top and the tagging results of the query set for 4 prompt combinations are shown at the bottom.

dataset $\mathcal{D}$. After training on the training dataset $\mathcal{D}_{\mathcal{S}}$, the model is required to predict labels for sentences from the test dataset $\mathcal{D}_{\mathcal{T}}$.

Different from the standard NER task, the few-shot NER setting consists of a meta training phase and a test phase. At the meta training phase, the model trains on a training dataset $\mathcal{D}_{\mathcal{S}}$. At the test phase, for various test datasets $\{\mathcal{D}_{\mathcal{T}}^{(j)}\}$, with only few labeled samples, the model is required to perform quick adaptions. In this paper, we mainly focus on two evaluation protocols and two task formulations which will be explained as follows.

3.2 Evaluation protocols

Following Ding et al. (2021); Ma et al. (2022a), we summarize two evaluation protocols as follows.

Episode Evaluation An episode, or a task, is defined as a pair of one support set and one query set $(\mathcal{S}, \mathcal{Q})$ each consisting of sentences downsampled from the test set. For an N-way K-shot downsampling scheme, there are N labels among the support set $\mathcal{S}$ where each label is associated with K examples. The query set $\mathcal{Q}$ shares the same label set with the support set. Based on the support set, the model is required to predict labels for the query set. To perform an episode evaluation, a collection of T episodes $\{(\mathcal{S}_t, \mathcal{Q}_t)\}_{t=1}^{T}$ are prepared. The evaluation results are computed within each episode and are averaged over all T episodes.

Low-resource Evaluation Different from the few-shot episode evaluation, low-resource evaluation aims to directly evaluate the model on the whole test set. For a test dataset $\mathcal{D}_{\mathcal{T}}$ with a label set $\mathcal{C}_{\mathcal{T}}$, a support set $\mathcal{S}$ associated with the labels from $\mathcal{C}_{\mathcal{T}}$ is constructed by K-shot downsampling such that each label has K examples in $\mathcal{S}$. Based on the support set $\mathcal{S}$, the model is required to predict labels for the query set which is the rest of the test set $\mathcal{D}_{\mathcal{T}}$. To perform a low-resource evaluation, T different runs of support set sampling are run and averaged.

3.3 Task formulation

Following Yang and Katiyar (2020), we formulate few-shot NER tasks in the following two ways.

Tag-Set Extension To mimic the scenario that new classes of entities emerge in some domain, Yang and Katiyar (2020) propose the tag-set extension formulation. Starting with a standard NER dataset $(\mathcal{D}_{train}, \mathcal{D}_{test})$ with label set $\mathcal{C}$, they split $\mathcal{C}$ into d parts, namely $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_d$. Then for each label split $\mathcal{C}_i$, a train set $\mathcal{D}_{train}^{(i)}$ is constructed from $\mathcal{D}_{train}$ by masking the labels in $\mathcal{C}_i$ to O (representing non-entities), and the corresponding test set $\mathcal{D}_{test}^{(i)}$ is constructed from $\mathcal{D}_{test}$ by masking the labels in $\mathcal{C} \setminus \mathcal{C}_i$ to O.

Domain Transfer Another task formulation is the domain transfer setting. Let $\mathcal{D}_{\mathcal{S}}$ be a training set of a standard NER task, and let $\{\mathcal{D}_{\mathcal{T}}{}^{(i)}\}$ be the test sets of standard NER tasks but from a different domain. The training set $\mathcal{D}_{\mathcal{S}}$ is referred to as a source domain, and the test sets $\{\mathcal{D}_{\mathcal{T}}{}^{(i)}\}$ constitute various target domains. In this setting, there may exist some overlapping entity classes between the source and target domains, but due to the domain gaps, it is still considered a few-shot setting.

Note that the task formulation is independent of the evaluation protocol, and different combinations will be considered in our experiments.

4 Method

4.1 Prompt Schemas

Motivated by existing prompt-based methods (Liu et al., 2021; Paolini et al., 2021) and the metric learning framework, our ProML provides label semantics by introducing prompts to metric learning models. We proposed a simple yet effective prompt class called the "mask-reducible prompts". Through this class of prompts, we can provide flexible prompts to the model which is consistent with metric learning methods that use token-level similarities as the metric. Starting with this schema, we will introduce two prompts that are used in ProML , the option-prefix prompt and the label-aware prompt.

4.2 Mask-Reducible Prompts

Suppose the raw input sequence is $\mathbf{x} = [x_1, x_2, \cdots, x_l]$. Let f_{prompt} be a prompt function mapping $\mathbf{x}$ to the prompted result $\mathbf{x}'$. We call this f_{prompt} is a mask-reducible prompt function if for all $\mathbf{x}$ and its prompted result $\mathbf{x}' = f_{prompt}(\mathbf{x})$, there exists a mask $\mathbf{m} \in [0, 1]^{|\mathbf{x}'|}$ such that $\mathbf{x}'[\mathbf{m} == 1] = \mathbf{x}$. Intuitively, this means there is only some insertions in the prompt construction so that we can revert $\mathbf{x}'$ back to $\mathbf{x}$ through a simple masking operation. The corresponding prompt of f_{prompt} is called a mask-reducible prompt.

Given a length preserving sequence-to-sequence encoder $Enc(\mathbf{x}; \theta)$, a sequence of input tokens $\mathbf{x}$, and a mask-reducible prompt function f_{prompt}, we first construct the prompted result $\mathbf{x}' = f_{prompt}(\mathbf{x})$, then pass the sequence $\mathbf{x}'$ through the encoder to get representations $\mathbf{h}' = Enc(\mathbf{x}'; \theta)$.

Since $Enc(\cdot; \theta)$ is length preserving, the length of $\mathbf{h}'$ is the same as $\mathbf{x}'$, and we can compute $\mathbf{h} = \mathbf{h}'[\mathbf{m} == 1]$ to get the representation for input tokens, where $\mathbf{m}$ is the desired mask that could reduce $\mathbf{x}'$ to $\mathbf{x}$ (i.e. $\mathbf{x}'[\mathbf{m} == 1] = \mathbf{x}$).

Through this process, the encoder receives the full prompts as its input while only the representations of raw input tokens are extracted.

Prompt A: Option Prefix Prompts An option prefix prompt takes the concatenation of all annotations as an option prefix to incorporate label semantics into modeling. Formally, for a given set of label options $\mathbf{S} = \{\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s}_{|\mathbf{S}|}\}$, we construct a mask-reducible prompting function $f_A(\mathbf{x}, \mathbf{S})$ associated with $\mathbf{S}$ using the template "$\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s}_{|\mathbf{S}|}$: $\mathbf{x}$". An example is given in Figure 2, where option prefix prompts reduce the label space to avoid incorrectly classify non-entities. The option prefix prompts inform the main model of which labels to predict, which can be used to learn label-dependent representations for computing the similarities.

Prompt B: Label-Aware Prompts A label-aware prompt appends the entity type to each entity occurrence in the input so that the model is aware of such information. While the aforementioned option prefix prompts incorporate global label information, the label-aware prompts introduce local information about each entity. Specifically, let $f_B(\mathbf{x}, \mathbf{y})$ be the prompt function. Given a sequence of input tokens $\mathbf{x}$ and its ground-truth label sequence $\mathbf{y}$, for each entity $\mathbf{e}$ that occurs in $\mathbf{x}$, we obtain its corresponding label $\mathbf{E}$ from the sequence $\mathbf{y}$, and replace $\mathbf{e}$ with an label-appended version "[$\mathbf{e}$|$\mathbf{E}$]" to construct the prompted result $\mathbf{x}' = f_B(\mathbf{x}, \mathbf{y})$. Both the entity $\mathbf{e}$ and its label $\mathbf{E}$ are sequences of tokens. Because the label-aware prompt can be applied when the ground-truth label is available, in our few-shot learning setting, we do not apply this prompt to the query set. An example is given in Figure 2, where label-aware prompts provide full label information in prompted inputs. More details will be explained in the following descriptions of our model architecture.

Note that it is possible to design other mask-reducible prompts for NER, which will be naturally handled by our framework. In our study, we find these two prompts work well practically and use them as instantiations to demonstrate the effectiveness of our framework.

4.3 Model and Training

The overall architecture of ProML is shown in Figure 1. Compared to the contrastive learning framework utilized by CONTaiNER (Das et al., 2022), our architecture uses a transformer backbone to encode different prompted inputs separately and employs a masked weighted average to obtain token representations, which will be elaborated as follows. These modifications significantly enhance the performance of our model when compared to the baseline method.

At the meta training phase, we sample mini-batches from the training set $\mathcal{D}_{train}$, where each mini-batch contains a few-shot episode $(\mathcal{S}_{train}, \mathcal{Q}_{train})$. We obtain the label set associated with the support set $\mathcal{S}_{train}$ and use a look-up dictionary to translate each label id to its natural language annotation. This leads to a set of label annotations $\mathbf{S}$. Then for an input sequence $\mathbf{x} = [x_1, x_2, \cdots, x_l]$ and its label sequence $\mathbf{y} = [y_1, y_2, \cdots, y_l]$ from the support set $\mathcal{S}_{train}$, we collect the prompted results $\mathbf{p_A} = f_A(\mathbf{x}, \mathbf{S})$, $\mathbf{p_B} = f_B(\mathbf{x}, \mathbf{y})$ and the corresponding masks $\mathbf{m_A}, \mathbf{m_B}$. These prompted results are then passed through a pretrained language model PLM. The average of outputs from the last four hidden layers are computed as the intermediate representations

$$\mathbf{h_A} = \mathrm{PLM}(\mathbf{p_A}), \mathbf{h_B} = \mathrm{PLM}(\mathbf{p_B}).$$

We perform a masked weighted average to obtain token representations

$$\mathbf{h} = \rho\mathbf{h_A}[\mathbf{m_A} == 1] + (1-\rho)\mathbf{h_B}[\mathbf{m_B} == 1],$$

where $\rho \in (0, 1)$ is a hyperparameter.

The token representations for the query set are computed similarly. However, during both training and testing, we only use the option-prefix prompt for the query set since the ground-truth label sequence will not be available at test time. As a result, we do not perform a weighted average for the query set. After obtaining the token representations, two projection layers f_μ, f_Σ are employed to produce two Gaussian embeddings, i.e., the mean and precision parameters of a d-dimensional Gaussian distribution $\mathcal{N}_{(\mu,\Sigma)}$ for each token in the query and support sets (Das et al., 2022).

Given the Gaussian embeddings for samples in both the support and query sets, we compute the distance metrics. Similar to CONTaiNER (Das et al., 2022), for a token x_i from the support set $\mathcal{S}_{train}$ and a token x'_j from the query set $\mathcal{Q}_{train}$, the distance between two tokens x_i, x'_j is defined as the Jenson-Shannon divergence (Fuglede and Topsøe, 2004) of their Gaussian embeddings, i.e.,

$$\begin{aligned} dist(x_i, x'_j) &= D_{JS}(\mathcal{N}_i, \mathcal{N}'_j) \\ &= \frac{1}{2}(D_{KL}(\mathcal{N}_{(\mu_i,\Sigma_i)}||\mathcal{N}_{(\mu'_j,\Sigma'_j)}) \\ &\quad + D_{KL}(\mathcal{N}_{(\mu'_j,\Sigma'_j)}||\mathcal{N}_{(\mu_i,\Sigma_i)})), \end{aligned}$$

where D_{KL} refers to the Kullback–Leibler divergence.

The similarity between x_i and x'_j is then defined as $s(x_i, x'_j) = \exp(-dist(x_i, x'_j))$. Let $\overline{\mathcal{S}}_{train}, \overline{\mathcal{Q}}_{train}$ be collections of all tokens from sentences in $\mathcal{S}_{train}, \mathcal{Q}_{train}$. For each $q \in \overline{\mathcal{Q}}_{train}$, the associated loss function is computed as

$$\ell(q) = -\log \frac{\sum_{p \in \mathcal{X}_q} s(q,p)/|\mathcal{X}_q|}{\sum_{p \in \overline{\mathcal{S}}_{train}} s(q,p)},$$

where $\mathcal{X}_q$ is defined by $\mathcal{X}_q = \{p \in \overline{\mathcal{S}}_{train} | p, q \text{ have the same labels}\}$. The overall loss function within a mini-batch is the summation of token-level losses, $L = \frac{1}{|\overline{\mathcal{Q}}_{train}|}\sum_{q \in \overline{\mathcal{Q}}_{train}} \ell(q)$.

4.4 Nearest Neighbor Inference

At test time, we compute the intermediate representations for tokens from the support and query sets just as we did during the meta training phase. Following CONTaiNER (Das et al., 2022), we no longer use the projection layers f_μ, f_Σ at test time but directly perform nearest neighbor inference using the token representations $\mathbf{h}$. For each query token, according to the Euclidean distance in the representation space, we compute the distance to each entity type by the distance to the nearest tokens from the support set associated with that entity type and assign the nearest entity type to the query token. For the k shot setting where $k > 1$, we also use the average distance of the nearest k neighbors associated with each entity type as the distance to the entity types.

Table 1: Evaluation results of ProML and 8 baseline methods in low-resource evaluation protocol for both tag-set extension and domain transfer tasks. Results with ⋆ are reported by the original paper, and those with † are reproduced in our experiments. We report the averaged micro-F1 score together with standard deviation. "Onto-A" denotes group-A set of OntoNotes dataset.

Method	Tag-Set Extension			Domain Transfer				Avg.
	Onto-A	Onto-B	Onto-C	CoNLL	WNUT	I2B2	GUM	
1-shot								
ProtoBERT(⋆)	19.3±3.9	22.7±8.9	18.9±7.9	49.9±8.6	17.4±4.9	13.4±3.0	17.8±3.5	22.77
NNShot(⋆)	28.5±9.2	27.3±12.3	21.4±9.7	61.2±10.4	22.7±7.4	15.3±1.6	10.5±2.9	26.7
StructShot(⋆)	30.5±12.3	28.8±11.2	20.8±9.9	62.4±10.5	24.2±8.0	21.4±3.8	7.8±2.1	27.99
CONTaiNER(⋆)	32.2±5.3	30.9±11.6	32.9±12.7	57.8±5.5	24.2±7.24	16.4±3.19	**17.9±2.28**	30.33
ProtoBERT(†)	8.39±2.16	17.12±4.04	8.4±1.94	53.09±9.89	21.17±4.71	15.85±4.89	11.91±3.01	19.42
NNShot(†)	21.97±7.11	33.89±7.1	21.73±6.78	59.76±8.63	26.53±4.54	15.0±3.63	10.33±3.08	27.03
StructShot(†)	24.02±6.24	36.42±8.22	22.70±6.65	60.84±7.62	29.16±4.88	18.34±2.70	11.17±2.18	28.95
CONTaiNER(†)	31.63±11.74	51.33±8.97	39.97±3.81	57.89±16.79	26.67±8.65	18.96±3.97	12.07±1.53	34.07
TransferBERT(†)	7.44±5.97	8.97±4.94	7.34±3.42	47.09±11.02	11.83±5.07	**35.25±4.21**	8.97±2.56	18.13
DualEncoder(†)	0.83±0.62	2.86±1.70	2.55±1.37	54.63±3.43	36.03±2.02	14.63±3.10	11.87±0.76	17.63
EntLM(†)	5.79±4.22	10.11±4.13	8.49±5.0	50.47±6.74	27.7±7.66	7.85±2.81	8.85±1.17	17.04
DemonstrateNER(†)	0.98±0.83	2.02±2.1	4.02±3.23	16.12±7.33	20.38±8.02	13.29±4.73	3.24±1.34	8.58
ProML	**37.94±6.08**	**53.74±3.6**	**46.27**±10.72	**69.16±4.47**	**43.89±2.17**	24.98±3.44	15.29±1.89	**41.61**
5-shot								
ProtoBERT(⋆)	30.5±3.5	38.7±5.6	41.1±3.3	61.3±9.1	22.8±4.5	17.9±1.8	19.5±3.4	33.11
NNShot(⋆)	44.0±2.1	51.6±5.9	47.6±2.8	74.1±2.3	27.3±5.4	22.0±1.5	15.9±1.8	40.36
StructShot(⋆)	47.5±4.0	53.0±7.9	48.7±2.7	74.8±2.4	30.4±6.5	30.3±2.1	13.3±1.3	42.57
CONTaiNER(⋆)	51.2±5.9	55.9±6.2	61.5±2.7	72.8±2.0	27.7±2.2	24.1±1.9	24.4±2.2	45.37
ProtoBERT(†)	25.81±3.0	31.49±4.6	32.08±2.12	65.76±5.34	32.81±8.78	35.05±12.25	25.02±2.66	35.43
NNShot(†)	39.49±5.96	50.18±4.99	45.98±4.61	70.79±3.44	33.68±5.21	29.50±2.89	19.04±2.38	41.24
StructShot(†)	35.68±6.17	51.30±4.61	47.85±4.74	71.23±3.62	35.36±2.99	27.08±3.17	19.67±2.45	41.17
CONTaiNER(†)	45.62±6.58	67.70±2.80	59.84±2.62	75.48±2.80	35.83±5.51	30.14±3.35	16.19±0.68	47.26
TransferBERT(†)	21.48±5.73	41.97±5.65	45.24±4.33	69.93±3.98	35.64±3.55	47.89±7.02	27.50±1.27	41.38
DualEncoder(†)	7.61±2.50	16.41±1.22	26.37±7.25	67.05±3.69	36.82±1.09	23.27±2.26	24.55±1.12	28.87
EntLM(†)	21.29±5.77	35.7±6.2	28.8±6.62	60.58±9.39	30.26±3.99	13.51±2.4	13.35±1.9	29.07
DemonstrateNER(†)	49.25±10.34	63.02±4.64	61.07±8.08	73.13±4.01	43.85±2.56	36.36±4.58	18.01±2.81	49.24
ProML	**52.46±5.71**	**69.69±2.19**	**67.58±3.25**	**79.16±4.49**	**53.41±2.39**	**58.21±3.58**	**36.99±1.49**	**59.64**

5 Experiments

5.1 Setup

Datasets We conduct experiments on multiple datasets across two few-shot NER formulations, tag-set extension and domain transfer. Following Das et al. (2022); Yang and Katiyar (2020), we split OntoNotes 5.0 (Weischedel et al., 2013) into Onto-A, Onto-B, and Onto-C for the tag-set extension formulation. For the domain transfer formulation, we use OntoNotes 5.0 (Weischedel et al., 2013) as the source domain, CoNLL'03 (Sang and Meulder, 2003), WNUT'17 (Derczynski et al., 2017), I2B2'14 (Stubbs and Uzuner, 2015), and GUM (Zeldes, 2017) as target domains. We also take Few-NERD (Ding et al., 2021) as one of the tag-set extension tasks, which is a large-scale human-annotated dataset speciallly designed for few-shot NER. The datasets statistics are presented in Table 3. We adopt the IO tagging scheme, where a label "O" is assigned to non-entity tokens and an entity type label is assigned to entity tokens. We also transform the abbreviated label annotations into plain texts; e.g., [LOC] to [location].

Baselines Our baselines include metric learning based methods such as the prototypical networks ProtoBERT (Snell et al., 2017; Fritzler et al., 2019; Hou et al., 2020), a nearest neighbor based network NNShot and its viterbi decoding variant StructShot (Yang and Katiyar, 2020), and a contrastive learning method CONTaiNER (Das et al., 2022). We also include a classification head based method TransferBERT (Hou et al., 2020) based on a pretrained BERT (Devlin et al., 2019). Existing method that make use of label semantics, DualEncoder (Ma et al., 2022a) is also reproduced for comparison. Recent prompt-based methods EntLM (Ma et al., 2021) and DemonstrateNER (Lee et al., 2022) are also employed as the baselines as well. We also compare our model with the recently-introduced based meth-

Table 2: Evaluation results of ProML and 7 baseline methods in episode evaluation protocol for FewNERD dataset. Results with ⋆ are reported by the original paper, and those with † are reproduced in our experiments. We report the averaged micro-F1 score together with standard deviation.

Method	1-shot		5-shot		Avg.
	INTRA	INTER	INTRA	INTER	
ProtoBERT(⋆)	20.76	38.83	42.54	58.79	40.23
NNShot(⋆)	25.78	47.24	36.18	55.64	41.21
StructShot(⋆)	30.21	51.88	38.00	57.32	44.35
CONTaiNER(⋆)	40.43	53.70	55.95	61.83	52.98
ESD(⋆)	36.08±1.60	59.29±1.25	52.14±1.50	69.06±0.80	54.14
DecomposedMetaNER(⋆)	49.48±0.85	64.75±0.35	62.92±0.57	71.49±0.47	62.16
ProtoBERT(†)	25.8±0.35	47.59±0.84	50.19±0.65	65.05±0.39	47.16
NNShot(†)	33.32±0.69	52.29±0.88	45.61±0.52	59.63±0.48	47.71
StructShot(†)	34.51±0.68	53.1±0.92	46.88±0.48	60.45±0.51	48.74
CONTaiNER(†)	37.12±1.01	55.19±0.43	49.22±0.34	62.64±0.33	51.04
TransferBERT(†)	22.43±1.49	38.26±2.36	48.95±1.23	62.2±1.36	42.96
ProML	**58.08±0.75**	**68.76±0.4**	**68.95±0.36**	**75.11±0.52**	**67.73**

Table 3: Statistics of Datasets

Dataset	**Domain**	**# Class**	**# Sample**
Few-NERD	Wikipedia	66	188K
OntoNotes	General	18	76K
CoNLL'03	News	4	20K
I2B2'14	Medical	23	140K
WNUT'17	Social	6	5K
GUM	Mixed	11	3.5K

ods DecomposeMetaNER (Ma et al., 2022b) and ESD (Wang et al., 2022). [2] For a fair comparison, we use bert-base-uncased (Devlin et al., 2019) as the PLM encoder and adopted the same pre-trained encoder in all the reproducible experiments of the baseline methods.

Evaluation Protocols Following Das et al. (2022); Yang and Katiyar (2020), we use the low-resource evaluation protocol for domain transfer tasks and for the tag-set extension tasks Onto-A, Onto-B, and Onto-C. Since Few-NERD (Ding et al., 2021) is specifically designed for episode evaluation, all of our experiments on Few-NERD dataset are evaluated under episode evaluation protocol. We follow the N-way K-shot downsampling setting proposed by Ding et al. (2021). For episode evaluation, we conduct 5 different runs of experiments, each of them contains 5000 test episodes. For low-resource evaluation, 10 different runs of support set sampling is performed.

5.2 Main Results

The main results of low-resource evaluation and episode evaluation are shown in Tables 1 and 2 respectively. Training details are provided in Appendix A.1. Our method achieves new state-of-the-art (SOTA) results under 16 out of the 18 considered settings. To compare with previous SOTA across different settings, we collect the relative improvement fractions from all settings and then compute an average and a maximum over these fractions. The result shows that ProML substantially outperforming the previous SOTA by an average of 9.12% and a maximum of 34.51% (from 28% to 37% on GUM 5-shot) in relative gains of micro F1. These outstanding results show that our method is effective for few-shot NER tasks.

The generalization difficulties are affected by both the label space and the domain gap. For example, Onto-A, B, and C datasets share the same domain but are constructed to have disjoint label space. CoNLL is a subset of the OntoNotes dataset, so its performance is much better than other domains.

Compared with the other baselines, the performances of prompt-based baselines decrease by a larger margin in the 1-shot settings since they heavily rely on finetuning on support sets.

5.3 Ablation Study and Analysis

The ablation study results for prompts choices and averaging weights on all tag-set extension tasks are shown in Table 4, 5. We adopt the episode

[2] The dataset we used is Few-NERD Arxiv V6 Version, while Ma et al. (2022b); Wang et al. (2022) reported their performances in the papers based on an earlier version (i.e. Arxiv V5 Version). We find the performances on the latest Few-NERD dataset on their official github repo at https://github.com/microsoft/vert-papers/tree/master/papers/DecomposedMetaNER.

Table 4: Ablation Study for ProML . The tuple indicates which prompts are used in the support set and query set. The variant **A, A** refers to using the option prefix prompt only in both the support set and query set. **plain+A** ($\rho = 0.5$)**, plain** refers to that the original inputs and option prefix prompts are used for the support set with an averaging weight $\rho = 0.5$, while the query set only use origin inputs. **A+B, A** is our ProML method.

Setting	Model	Onto-A	Onto-B	Onto-C	INTRA	INTER
5-shot	plain, plain	42.1±1.03	62.87±0.52	50.58±0.98	53.08±0.85	65.66±0.08
	A, A	47.04±1.01	65.42±0.62	55.77±1.19	66.19±0.72	73.9±0.34
	B, plain	39.58±2.26	51.17±1.01	40.28±3.55	49.9±1.68	65.31±1.36
	plain+A ($\rho = 0.3$), plain	40.43±1.64	62.41±1.3	49.51±2.78	56.4±1.02	68.15±0.42
	plain+A ($\rho = 0.5$), plain	42.35±1.32	64.37±0.48	51.94±1.06	56.69±0.93	68.73±0.25
	plain+A ($\rho = 0.7$), plain	42.75±2.18	64.52±0.57	53.07±1.79	55.33±1.34	68.37±0.26
	plain+B ($\rho = 0.3$), plain	46.85±1.32	58.0±1.68	50.54±1.71	54.18±1.25	67.03±0.7
	plain+B ($\rho = 0.5$), plain	52.34±0.31	62.07±2.15	55.9±0.5	57.75±0.32	68.22±0.25
	plain+B ($\rho = 0.7$), plain	52.37±0.57	66.39±1.22	57.7±0.71	57.52±0.81	69.04±0.2
	A+B ($\rho = 0.3$), A	52.76±0.82	59.34±1.49	55.52±0.89	66.95±0.82	73.51±0.3
	A+B ($\rho = 0.5$), A	55.29±0.98	62.49±1.2	59.99±0.99	68.41±0.27	74.52±0.44
	A+B ($\rho = 0.7$), A	**55.76±1.06**	**67.09±0.49**	**62.57±0.47**	**68.95±0.36**	**75.11±0.52**

evaluation protocol due to its low variance. More ablations and the training curve, case study are placed in Appendix A.3, A.2, A.4, respectively.

Option Prefix Prompts & Label-Aware Prompts According to Table 4, overall, by comparing the best variant of prompting methods to "plain", using prompting consistently outperforms the methods without prompting. The improvements are consistent with our motivation in the earlier sections. With the help of label semantic annotations, the model is able to leverage this information to better learn the representation of each token. In addition, the model does not need to spend much capacity memorizing and inferring the underlying entity types for input tokens, which is crucial in the few-shot setting where labels are scarce.

The performance of variant "B, plain" is not good since only the support set leverages label-aware prompts so that there is a gap between the amounts of additional information from support to query. Thus there is a potential risk that the model only emphasizes these labels in support inputs while neglecting the semantics for tokens themselves, causing an overfitting problem. However, after introducing a weighted average, as shown in "plain+B, plain", the performance significantly improves. This observation suggests that the label-aware prompt is useful and the weighted average mitigates the overfitting by reducing the gaps between support and query.

As we will show in the next section, combining the two prompts always leads to the best performance because the model is able to dynamically adapt to the two representations.

Effect of Masked Weighted Average As reported before, a weighted average could reduce the gaps between computing representations for the support set and the query set and make use of the information provided by label-aware prompts. By adjusting the averaging weight ρ, we are able to balance the weights of the two representations for different data distributions.

We compared different averaging settings in 4. The option prefix only variant "A, A" performs better than "plain+A, plain" because the label option information is provided to both support and query. The performance of "plain+B, plain" and "A+B, A" improve as ρ increases, which is consistent with our motivation

According to Table 4, with a properly selected averaging weight ρ, our ProML outperforms all baselines by a large margin among all tested datasets, which indicates that both prompts contribute to our final performance. Importantly, $\rho = 0.7$ tends to work well in most of the settings, which can be used as the default hyperparameter in our framework without tuning.

Visualizing Embedding Space We visualize the token representations from support sets and query sets over several episodes from the test set of Few-NERD INTRA, as Figure 3 shows. We observe that the token representations produced by ProML are concentrated in different clusters. In addition, we shall observe a clear decision boundary between different clusters. On the contrary, CONTaiNER seems to learn scattered, less separable features.

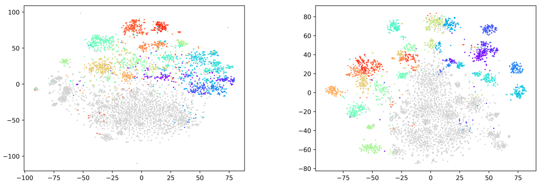


Figure 3: TSNE visualization of token representations under the Few-NERD test set for CONTaiNER (on the left) and ProML (on the right), where each color represents an entity type (grey for non-entities). We only keep a fraction of 20% among the non-entities to make the TSNE visualization clearer.

6 Conclusions

We propose a novel prompt-based metric learning framework ProML for few-shot NER that leverages multiple prompts to guide the model with label semantics. ProML is a general framework consistent with any token-level metric learning method and can be easily plugged into previous methods. We test ProML under 18 settings and find it substantially outperforms previous SOTA results by an average of 9.12% and a maximum of 34.51% in relative gains of micro F1. We perform ablation studies to show that multiple prompt schemas benefit the generalization ability for our model. We demonstrate the visualization results for embedding space to unseen entities, showing that comparing with previous SOTA, ProML learns better representations. We also present case studies and perform some analysis.

7 Limitations

Although we discussed different task formulations and evaluation protocols, the few-shot settings are simulated by downsampling according to existing works, which is slightly different from the real scenario.

References

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2019. Few-shot text classification with distributional signatures. arXiv preprint arXiv:1908.06039.

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.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017, pages 140–147. 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 Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3198–3213, Online. Association for Computational Linguistics.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In SAC, pages 993–1000. ACM.

Bent Fuglede and Flemming Topsøe. 2004. Jensen-shannon divergence and hilbert space embedding. In Proceedings of the 2004 IEEE International Symposium on Information Theory, ISIT 2004, Chicago Downtown Marriott, Chicago, Illinois, USA, June 27 - July 2, 2004, page 31. IEEE.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. arXiv preprint arXiv:1902.10482.

Abbas Ghaddar and Philippe Langlais. 2017. Winer: A wikipedia annotated corpus for named entity recognition. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 413–422.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

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. arXiv preprint arXiv:1810.10147.

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 Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1381–1393, Online. Association for Computational Linguistics.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. arXiv preprint arXiv:2012.14978.

Dong-Ho Lee, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2022. Good examples make a faster learner: Simple demonstration-based learning for low-resource NER. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2687–2700, Dublin, Ireland. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022a. Label semantics for few shot named entity recognition. In Findings of the Association for Computational Linguistics: ACL 2022, pages 1956–1971, Dublin, Ireland. Association for Computational Linguistics.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. Template-free prompt tuning for few-shot ner. arXiv preprint arXiv:2109.13532.

Tingting Ma, Huiqiang Jiang, Qianhui Wu, Tiejun Zhao, and Chin-Yew Lin. 2022b. Decomposed meta-learning for few-shot named entity recognition. In FINDINGS.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. arXiv preprint arXiv:2101.05779.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003, pages 142–147. ACL.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. CoRR, abs/2009.07118.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In NIPS, pages 4077–4087.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. J. Biomed. Informatics, 58:S20–S29.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. Advances in neural information processing systems, 29.

Peiyi Wang, Runxin Xu, Tianyu Liu, Qingyu Zhou, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022. An enhanced span-based decomposition method for few-shot sequence labeling. ArXiv, abs/2109.13023.

Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2021. Meta self-training for few-shot neural sequence labeling. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pages 1737–1747.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. Linguistic Data Consortium, Philadelphia, PA, 23.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6365–6375, Online. Association for Computational Linguistics.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In International Conference on Machine Learning, pages 7115–7123. PMLR.

Amir Zeldes. 2017. The GUM corpus: creating multilayer resources in the classroom. Lang. Resour. Evaluation, 51(3):581–612.

A Appendix

A.1 Training Details

We use AdamW (Loshchilov and Hutter, 2019) for optimization and the learning rate is set to 3×10^{-5}, linearly warming up during first 10% of all 10^4 training iterations. We use bert-base-uncased (Devlin et al., 2019) as the PLM encoder. The weight decay is set to 0.01 for all parameters of the model except the biases and layer norm layers. The value of hyperparameter ρ is chosen from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and is set to 0.7 by default (which is good enough for almost all cases). For fair comparison, we use the same Gaussian embedding dimension $d = 128$ as CONTaiNER (Das et al., 2022). A single experiment run takes about 1 hour on a single RTX3090.

A.2 Training Curve

Our architecture of using multiple prompts also mitigates overfitting. We conduct two experiments on Few-NERD to prove this empirically. Figure 4 demonstrates the training curves for CONTaiNER (Das et al., 2022) and our model. From the curves we can see that the trends of performances over training set are similar while the performance of CONTaiNER on dev set stops increasing much earlier than ProML . Compared with CONTaiNER, our model gets much better in the later epochs. This shows that ProML suffers less from overfitting in the few-shot setting.

A.3 Ablations

Ablation Table for both 1-shot and 5-shot Due to page limit, we leave ablation for 1-shot to the appendix. The full version is in Table 5.

Ablation for Replacing Labels with Noises & Removing Separators We made an experiment to replace labels with random strings (both in train and test, same entity type shares same label) to show the effect of label semantics. According to Table 6, the results from "ProML noise-label" are significantly worse than our ProML, but still comparable with the previous SOTA on Few-NERD dataset. This shows that the semantics of the label really help and label-aware prompts can provide useful information even if the labels are noisy. We also made an abbreviation for the selection of separator. In the experiment "ProML no-sep" from Table 6 where all separators were removed, the performances drops to some extent but there is no significant difference.

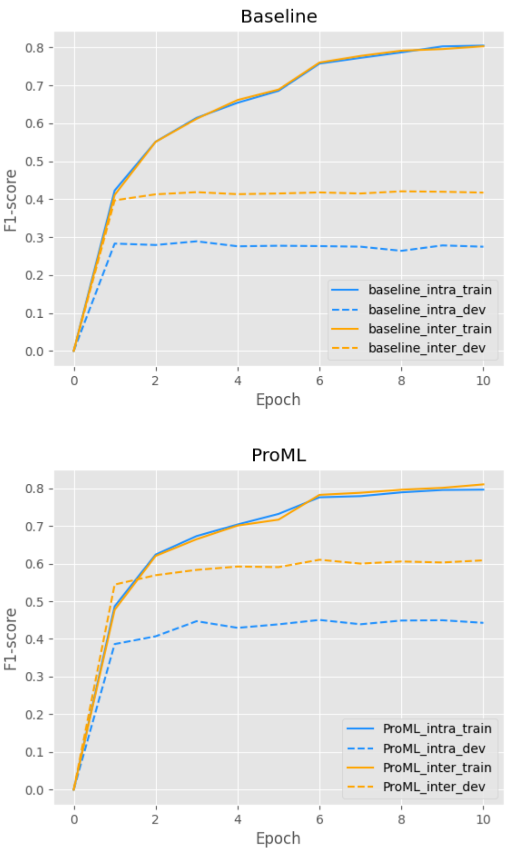


Figure 4: Training curves for CONTaiNER (Das et al., 2022) baseline (on the left) and our model (on the right). The experiments are conducted under Few-NERD INTRA 1-shot and INTER 1-shot setting.

A.4 Case Study

We present several randomly-selected cases from ProML and CONTaiNER using the test-set results of WNUT 1-shot domain transfer task. The results are in Table 7. We can see that ProML gives better predictions than CONTaiNER (Das et al., 2022) for most cases. Specifically, CONTaiNER often misses entities or incorrectly classifies non-entities.

Table 5: Ablation Study for ProML (1-shot and 5-shot). The tuple indicates which prompts are used in the support set and query set. The variant **A, A** refers to using the option prefix prompt only in both the support set and query set. **plain+A** ($\rho = 0.5$**), plain** refers to that the original inputs and option prefix prompts are used for the support set with an averaging weight $\rho = 0.5$, while the query set only use origin inputs. **A+B, A** is our ProML method. All results in this table are produced by the episode evaluation protocol.

Setting	Model	Onto-A	Onto-B	Onto-C	INTRA	INTER
1-shot	plain, plain	27.4±0.93	49.91±1.22	32.51±0.98	37.17±0.98	54.11±0.72
	A, A	30.99±0.91	52.57±0.82	37.44±1.3	51.0±0.74	65.86±0.56
	B, plain	25.8±0.89	34.76±3.3	24.02±1.1	37.99±2.55	58.98±1.57
	plain+A ($\rho = 0.3$), plain	29.79±1.79	50.43±1.04	33.51±1.86	43.35±1.0	59.95±0.35
	plain+A ($\rho = 0.5$), plain	30.81±1.41	50.51±0.83	34.8±1.05	43.25±0.54	60.06±0.49
	plain+A ($\rho = 0.7$), plain	28.32±1.37	50.79±0.87	34.27±0.92	41.42±0.55	59.1±0.5
	plain+B ($\rho = 0.3$), plain	31.03±0.91	40.39±1.5	31.67±1.8	45.16±0.39	62.27±0.63
	plain+B ($\rho = 0.5$), plain	33.58±0.44	45.11±0.85	36.25±0.93	45.1±0.41	62.67±0.78
	plain+B ($\rho = 0.7$), plain	33.42±0.46	49.44±0.96	38.67±0.61	43.07±0.44	61.09±0.5
	A+B ($\rho = 0.3$), A	33.43±1.42	42.07±1.49	35.26±1.1	57.16±1.52	68.04±0.82
	A+B ($\rho = 0.5$), A	33.31±0.57	42.94±2.1	39.27±0.52	**58.08±0.75**	68.43±0.6
	A+B ($\rho = 0.7$), A	**35.58±0.4**	**50.53±1.03**	**42.12±0.84**	57.19±0.91	**68.76±0.4**
5-shot	plain, plain	42.1±1.03	62.87±0.52	50.58±0.98	53.08±0.85	65.66±0.08
	A, A	47.04±1.01	65.42±0.62	55.77±1.19	66.19±0.72	73.9±0.34
	B, plain	39.58±2.26	51.17±1.01	40.28±3.55	49.9±1.68	65.31±1.36
	plain+A ($\rho = 0.3$), plain	40.43±1.64	62.41±1.3	49.51±2.78	56.4±1.02	68.15±0.42
	plain+A ($\rho = 0.5$), plain	42.35±1.32	64.37±0.48	51.94±1.06	56.69±0.93	68.73±0.25
	plain+A ($\rho = 0.7$), plain	42.75±2.18	64.52±0.57	53.07±1.79	55.33±1.34	68.37±0.26
	plain+B ($\rho = 0.3$), plain	46.85±1.32	58.0±1.68	50.54±1.71	54.18±1.25	67.03±0.7
	plain+B ($\rho = 0.5$), plain	52.34±0.31	62.07±2.15	55.9±0.5	57.75±0.32	68.22±0.25
	plain+B ($\rho = 0.7$), plain	52.37±0.57	66.39±1.22	57.7±0.71	57.52±0.81	69.04±0.2
	A+B ($\rho = 0.3$), A	52.76±0.82	59.34±1.49	55.52±0.89	66.95±0.82	73.51±0.3
	A+B ($\rho = 0.5$), A	55.29±0.98	62.49±1.2	59.99±0.99	68.41±0.27	74.52±0.44
	A+B ($\rho = 0.7$), A	**55.76±1.06**	**67.09±0.49**	**62.57±0.47**	**68.95±0.36**	**75.11±0.52**

Table 6: Ablations for removing separators in prompts and replacing labels with random noises. All methods are evaluated in episode evaluation protocol for Few-NERD dataset.

Method	1-shot		5-shot		Avg.
	INTRA	INTER	INTRA	INTER	
ProML	**58.08±0.75**	**68.76±0.4**	**68.95±0.36**	**75.11±0.52**	**67.73**
ProML no-sep	55.66±0.75	68.03±0.27	67.82±0.17	74.82±0.32	66.58
ProML noise-label	51.99±0.84	65.8±0.69	62.09±0.44	72.5±0.43	63.10

Table 7: Case study: An illustration of some cases from the WNUT test set. There are 6 entities: person (PER), location (LOC), product (PRO), creative work (CW), miscellaneous (MIS), group (GRO). Here blue color represents correct predictions, while red color represents mistakes.

GroundTruth	**ProML**	**CONTaiNER**
wow emma$_{PER}$ and kaite$_{PER}$ is so very cute and so funny i wish im ryan$_{PER}$	wow emma$_{PER}$ and kaite$_{PER}$ is so very cute and so funny i wish im ryan$_{PER}$	wow emma$_{PER}$ and kaite$_{PER}$ is so very cute and so funny i wish im$_{PER}$ ryan$_{PER}$
these trap came from taiwan$_{LOC}$.	these trap came from taiwan$_{LOC}$.	these trap came from taiwan$_{LOC}$.
great video ! good comparisons between the ipad$_{PRO}$ and the ipad$_{PRO}$ pro$_{PRO}$!	great video ! good comparisons between the ipad$_{PRO}$ and the$_{PRO}$ ipad$_{PRO}$ pro$_{PRO}$!	great video ! good comparisons between the ipad and the ipad pro$_{PRO}$!
thanks for colors superheroes kids videos !) like learn$_{CW}$ colors$_{CW}$ and$_{CW}$ numbers$_{CW}$!)	thanks for colors$_{CW}$ superheroes$_{CW}$ kids videos !) like learn$_{CW}$ colors$_{CW}$ and$_{CW}$ numbers$_{CW}$!)	thanks for colors$_{COR}$ superheroes kids videos !) like learn colors and numbers !)
i pronounce it nye-on cat	i pronounce it nye-on cat	i pronounce it nye-on$_{PRO}$ cat$_{PRO}$

ACL 2023 Responsible NLP Checklist

A For every submission:

□ A1. Did you describe the limitations of your work?
Left blank.

□ A2. Did you discuss any potential risks of your work?
Left blank.

□ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

□ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B □ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C □ Did you run computational experiments?

Left blank.

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

□ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D □ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.