Labels Need Prompts Too: Mask Matching for Natural Language Understanding Tasks

Bo Li[1,2], **Wei Ye**[1*], **Quansen Wang**[3], **Wen Zhao**[1], **Shikun Zhang**[1]
[1]National Engineering Research Center for Software Engineering, Peking University
[2]School of Software and Microelectronics, Peking University
[3]Boston University
deepblue.lb@stu.pku.edu.cn, wye@pku.edu.cn,
quansenw@bu.edu, {zhaowen, zhangsk}@pku.edu.cn

Abstract

Textual label names (descriptions) are typically semantically rich in many natural language understanding (NLU) tasks. In this paper, we incorporate the prompting methodology, which is widely used to enrich model input, into the label side for the first time. Specifically, we propose a Mask Matching method, which equips an input with a prompt and its label with another, and then makes predictions by matching their mask representations. We evaluate our method extensively on 8 NLU tasks with 14 datasets. The experimental results show that Mask Matching significantly outperforms its counterparts of fine-tuning and conventional prompt-tuning, setting up state-of-the-art performances in several datasets. Mask Matching is particularly good at handling NLU tasks with large label counts and informative label names. As pioneering efforts that investigate the label-side prompt, we also discuss open issues for future study.

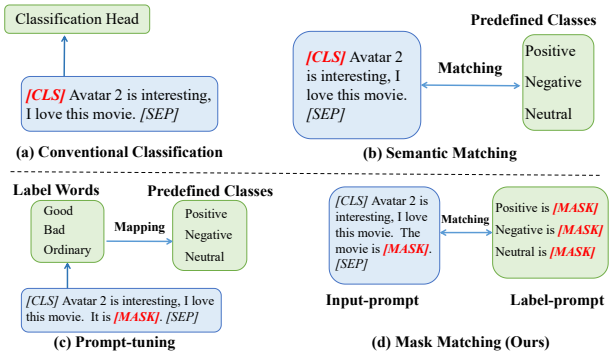


Figure 1: Conceptual illustration of our Mask Matching method and other popular paradigms. We use sentiment analysis as the example task.

1 Introduction

Large-scale pre-trained language models (PLMs) such as BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019) have achieved impressive performances on a wide range of natural language understanding (NLU) tasks, e.g., topic classification (Xu, Liu, and Abbasnejad 2022; Wu et al. 2022), sentiment analysis (Zhang et al. 2022), information extraction (Li et al. 2020; Lu et al. 2022), natural language inference (Dawkins 2021; Nighojkar and Licato 2021), and stance detection (Liu et al. 2021c; Jiang et al. 2022b).

In general, fine-tuning PLMs with a classification head on the downstream dataset is the dominant solution for most NLU tasks, as shown in Figure 1 (a). While this paradigm achieves impressive performances, it can not utilize textual semantics implied in label descriptions, which, however, have been proven to be beneficial for many downstream tasks (Xu, Liu, and Abbasnejad 2022; del Arco, Valdivia, and Klinger 2022; Jiang et al. 2022b; Liang et al. 2022; Obeidat et al. 2019; Huang et al. 2022; Sainz et al. 2021; Li et al. 2022). In this case, two other paradigms of NLU tasks can come to the rescue to some extent.

Semantic matching (Sainz et al. 2021; Huang et al. 2022; del Arco, Valdivia, and Klinger 2022) involves generating representations of inputs and labels, and making predictions based on their semantic distances, as illustrated in Figure 1 (b). This paradigm can naturally exploit semantic information from labels, e.g., utilizing representations of label names generated by PLMs. However, this paradigm heavily relies on label representations, and conventional modeling (e.g., with max-pooling or average-pooling) of label-related texts might not yield optimal ones.

Prompt-tuning (Liu et al. 2021a; Han et al. 2021; Chen et al. 2022) naturally exploits textual semantics of labels by manually designing proper label verbalizer (Schick and Schütze 2021; Gao, Fisch, and Chen 2021; Lee et al. 2022), as shown in Figure 1 (c). Nevertheless, selecting label words is non-trivial and labor-intensive for many tasks, such as entity typing (Ding et al. 2021b) and relation extraction (Zhang et al. 2017). Some researchers explore utilizing trainable virtual label words instead (Wang et al. 2022b; Chen et al. 2022; Han et al. 2021; Park et al. 2022). Though avoiding cumbersome verbalizer engineering, they completely discard label-related text information.

In this research, we present a new paradigm called Mask Matching that mines label semantic information using the prompting methodology on the label side. To capture the semantics of inputs and labels, we introduce a *label-prompt* in addition to the *input-prompt*, which results in effective predictions based on a simple matching strategy. In this way, the

* Corresponding author.

semantics of inputs and labels are captured by the mask representations from their corresponding prompts. Mask Matching combines the merits of traditional semantic matching and prompt-tuning paradigms and avoids the verbalizer engineering, yielding a conceptually simple and easy-to-implement method.

To verify the effectiveness of Mask Matching, we conduct extensive experiments on 8 NLU tasks across 14 datasets. Our method demonstrated remarkable performances on both full training setting (§5.1) and low-resource settings (§5.2) compared to its counterparts of fine-tuning and prompt-tuning. Additionally, it achieves comparable or better results than many state-of-the-art methods. Notably, Mask Matching exhibits more evident superiority when the predefined class in the datasets is numerous, and their names are informative. Besides the performance improvements, we also discuss the potential research directions upon Mask Matching (§6). Below we summarize our main contributions:

- We propose Mask Matching, a new natural language understanding (NLU) paradigm that simultaneously performs prompting on inputs and labels. It can be easily and effectively applied in most NLU tasks by matching the two mask representations of both sides.
- Extensive experiments show that Mask Matching significantly outperforms its counterparts of fine-tuning and prompt-tuning, and achieves competitive results compared with recent state-of-the-art models.
- As pioneering efforts that investigate the label-side prompt, we propose many open problems to inspire future studies in this direction.

2 Related Work

As this work aims to explore utilizing semantic information in label names, we briefly introduce how existing approaches utilize label semantic information.

Label semantics is beneficial to many NLU tasks, such as text classification (Xu, Liu, and Abbasnejad 2022; del Arco, Valdivia, and Klinger 2022), stance detection (Jiang et al. 2022b; Liang et al. 2022), named entity recognition (Obeidat et al. 2019; Huang et al. 2022) and relation classification (Sainz et al. 2021; Li et al. 2022). In recent works, researchers proposed several approaches to fully use the available semantic information in labels and achieve desirable performances, e.g., semantic matching methods and prompt-tuning methods.

The semantic matching method is the default solution for sentence-pair tasks, such as natural language inference (Rajpurkar et al. 2016) and paraphrase (Xu, Callison-Burch, and Dolan 2015). This method could utilize the label semantic via encoding label descriptions and achieves good results on tasks where labels contain rich semantic information (Wang et al. 2021; Sainz et al. 2021; Huang et al. 2022; del Arco, Valdivia, and Klinger 2022). Typically, Semantic matching method usually jointly encodes premise/hypothesis or input/labels and evaluates the relationship between both ends. This paradigm heavily relies on label representations, while conventional modeling (e.g., with max-pooling or average-pooling) of label-related texts might not yield optimal ones.

Prompt-tuning is an emerging paradigm in recent years. It could bridge the gap between pre-training and fine-tuning, showing surprising power on a wide range of NLP tasks (Liu et al. 2021a; Han et al. 2021; Chen et al. 2022). Specially, with a carefully designed template, prompt-tuning transforms the target task to a cloze style format and outputs the prediction via a special mask token. Prompt-tuning naturally utilizes label semantic information by manually designing properly label verbalizer (Schick and Schütze 2021; Gao, Fisch, and Chen 2021; Lee et al. 2022), such as “good” for "positive" and “bad” for "negative". To avoid human involvement, some researchers also explore utilizing trainable virtual label words (Wang et al. 2022b; Chen et al. 2022; Han et al. 2021; Park et al. 2022). Although most prompt-tuning methods perform well in low-resource scenarios (Liu et al. 2021c; Wang et al. 2022a; Liu, Chen, and Xu 2022), they still struggles to achieve on pair results compared with fine-tuning, especially when PLMs are relatively small and the training data is sufficient(Lester, Al-Rfou, and Constant 2021; Gao, Fisch, and Chen 2021; Zhong et al. 2022).

3 Approach

In this section, we first briefly introduce the preliminaries and the prompt-tuning method, then present our Mask Matching in detail.

3.1 Prompt-tuning

Unlike traditional fine-tuning methods that utilize the *[CLS]* token for NLU tasks (Devlin et al. 2019; Liu et al. 2019), prompt-tuning methods use a special mask token and a pre-defined template for prediction output. For instance, when dealing with a sentiment analysis task with input $[X]$, researchers may use the following template: *"[X]. It is [MASK]"*. Where the mask representation M_I is then converted to a class prediction by a predefined label verbalizer, e.g., {"good" $\rightarrow$ "positive", "bad" $\rightarrow$ "negative" and "ordinary" $\rightarrow$ "natural" }. In the above example, "good", "bad" and "ordinary" are label words, and "positive", "negative" and "natural" are label names. An example of such a system can be seen in Figure 1 (c). With a well-designed template and label verbalizer, prompt-tuning is effective in solving single-input tasks such as sentiment analysis and topic classification.

Despite the above tasks, prompt-tuning could also handle paired-input tasks by concatenating two inputs with a prompt. We take the paraphrase task as an example. Given two sentences $[X_1]$ and $[X_2]$, the prompt-tuning method will recompose inputs as $[\hat{X}]$: $[X_1]$ $[X_2]$ *The relation between two sentences is [MASK]*,

Depending on the task, we can employ either manually chosen real words (Schick and Schütze 2021; Gao, Fisch, and Chen 2021; Han et al. 2021; Lee et al. 2022) or virtual label tokens that can be trained (Wang et al. 2022b; Chen et al. 2022; Han et al. 2021; Park et al. 2022) as our label verbalizer. In this study, we opt for the latter, virtual token approach because it does not entail manual effort and obtains more consistent and better performance, particularly when the quantity of training data is substantial. To clarify, we refer

to the aforementioned prompts as *input-prompts* since they all exist on the input side.

3.2 Mask Matching

This section outlines Mask Matching, a new method that utilizes two mask tokens to learn more useful information from both the input and label sides. Overall, in addition to the *input-prompt* P_I mentioned in §3.1, Mask Matching involves a *label-prompt* P_L. For each label, *label-prompt* adds a template with a mask token after the label name. The mask representation M_L from P_L is used as the label representation. At the training phase, Mask Matching optimizes parameters by computing the cross-entropy loss between M_I and M_L over all predefined labels. Note that we use the same PLM to encode inputs and labels.

The *label-prompt* in Mask Matching eliminates the need for label verbalizer construction and enhances the utilization of semantic information in label names. The subsequent section explains how to employ Mask Matching for various NLU tasks. Based on the input format, we divide the NLU tasks examined in this research into two categories: **Single-input Tasks** and **Paired-input Tasks**.

Single-input Tasks. This type contains a wide range of NLU tasks with a single input, such as topic classification, sentiment analysis, entity typing, and relation classification. In these tasks, Mask Matching incorporates prompts on both the input and label sides simultaneously. As shown in Figure 2 (a), we use slightly different *input-prompt* variations for different tasks, and the *label-prompt* P_L is consistent across all tasks, defined as P_L = *is [MASK].*

- **Topic Classification and Sentiment Analysis.** In these tasks, we need to classify which class the input text $[X]$ belongs to. We set the *input-prompt* P_I = *It is [MASK]*, as shown in Figure 2 (a).
- **Entity Typing.** This task requires identifying the entity type for a given target entity. As shown in Figure 2 (a), the input text *[X] = Currently Ritek is the largest producer of OLEDs in the world*, and the target entity is *Ritek*. We use the following simple template in the *input-prompt*, where P_I = *The type of [target entity] is [MASK].*
- **Relation Classification.** This task aims to classify the relation between *head entity* and *tail entity* in a given input. As shown in Figure 2 (a), the input text *[X] = He was an army of the Korean War*, and the *head entity* and *tail entity* are *He* and *army*. The *input-prompt* P_I = *The relation between [head entity] and [tail entity] is [MASK]*. Note that we also add entity markers and entity type information to the input, as previous works show these two types of information could bring huge improvements (Soares et al. 2019; Tian et al. 2021).

Paired-input Tasks. These tasks need to identify the relationship between two given text pieces, e.g., neutral language inference, paraphrase, word in context and stance detection. For the input side, we concatenate two inputs with a task-specific *input-prompt* P_I. As for the label side, we use the same *label-prompt* P_L as described in Single-input Tasks.

- **Natural Language Inference and Paraphrase.** The above two tasks aim to distinguish the relationship between two sentences. We first concatenate two given sentences *[X_1]* and *[X_2]*, then add a simple *input-prompt* P_I, where P_I = *The relation between two sentences is [MASK].* An example is shown in Figure 2 (b).
- **Word in Context.** This is a semantic distinction task with paired-input. In this task, we have two input texts and two keywords, *[K_1]* and *[K_2]*, often the same word, but in different tenses or morphemes. The objective is to determine if the two keywords have similar meanings. For example, given the inputs *[X_1] = You must carry your camping gear* and *[X_2] = Sound carries well over water*, with *carry* and *carries* as the two keywords, the *input-prompt* P_I is added to the concatenated input, where P_I = *[K_1] is [MASK] to [K_2]*. This task is demonstrated in Figure 2 (b).
- **Stance Detection.** Here, we are asked to identify whether a text is in favor of, against, or neutral to a given target (e.g., an event, or a claim). As shown in Figure 2 (b), the given input text is *[X] = We are so becoming a failing nation. Between the rights of illegals and uneducated and now obese are claiming rights.* The target phrase is *[T] = illegal immigrant*. For the given input *[X]* and the target *[T]*, we add a *input-prompt* P_I after *[X]*, where P_I = *The stance of [T] is [MASK].*

Note that the label names of these tasks are all meaningful words or phrases. Thus we can directly use the *label-prompt* to learn useful semantic information from label names.

3.3 Training and Testing

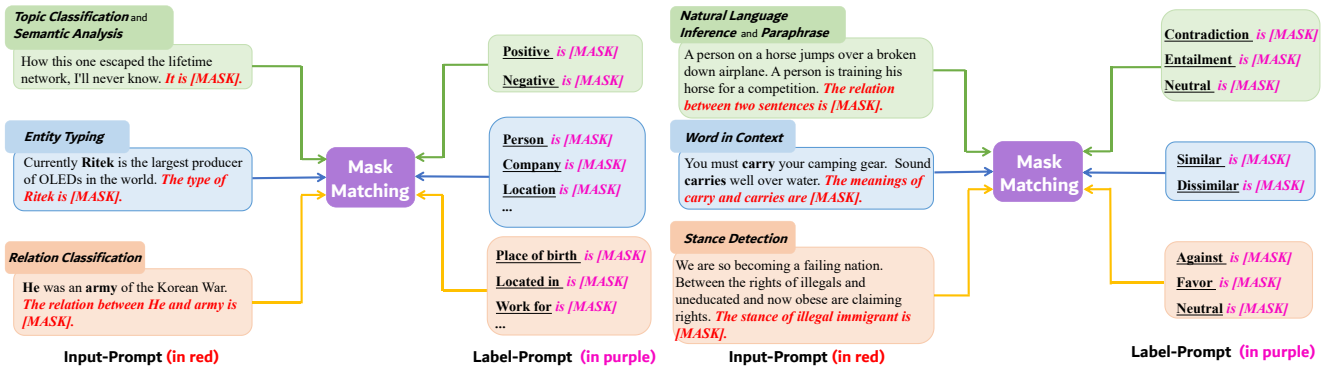
During the training phase, for a given sample with m predefined classes, we utilize the mask within the *input-prompt* as the input representation and m corresponding *[MASK]s* from the *label-prompts* as the label embeddings. We compute the cross-entropy loss between the input representation and label embeddings to compute the loss and optimize the entire PLM. For instance, let's consider the entity typing task, as illustrated in Figure 2(a). Assuming that the *[MASK]* representation from the input prompt is denoted as h, and the *[MASK]* representations from the label prompt are denoted as m_1, m_2, ..., m_n, where n represents the number of predefined labels. To begin, we derive the prediction probability using a softmax activation function as follows:

$$p = softmax(h \cdot [m_1, m_2, ..., m_n]^T), \tag{1}$$

where [*] means concatenation. Subsequently, we optimize the *cross-entropy (CE) loss* between *p* and the ground-truth label *q* (one-hot format):

$$loss = CE(p, q), \tag{2}$$

During testing, we calculate the dot product between the input representation and all label embeddings to generate the final prediction.



(a) Single-Input Tasks (b) Paired-Input Tasks

Figure 2: These examples provide an intuitive understanding of several single-input and paired-input tasks, all of which are extracted from real-world datasets. The highlighted bold target entities, keywords, and phrases to be identified, and the corresponding labels are underlined. The *input-prompts* are distinguished in red, while the *label-prompts* are showcased in purple. The mask in each *label-prompt* symbolizes the corresponding label. The cross-entropy loss in Mask Matching is calculated between the mask in the *input-prompt* and all masks in the *label-prompts*. It's important to note that the entire PLM is trainable. Better viewed in color.

4 Experimental Setup

4.1 Datasets

We use 8 different natural language understanding tasks across 14 datasets to verify the effectiveness of our proposed method. The metric and class numbers for each dataset are shown in Table 1. These datasets are chosen from a wide range of common NLU tasks. Some of them are selected from GLUE benchmark (Wang et al. 2019b) and SuperGLUE (Wang et al. 2019a), others are popular in various specific research fields, such as entity typing, relation classification, and stance detection. The first category is *single-input task*, such as topic classification (R8 and R52[1], we use the data split proposed by (Lin et al. 2021)), sentiment analysis (MR (Pang and Lee 2005) and IMDb (Maas et al. 2011)), entity typing (FEW-NERD (Ding et al. 2021b) and BBN (Weischedel and Brunstein 2005; Huang, Meng, and Han 2022)) and relation classification (TACRED (Zhang et al. 2017) and TACRED-Revisited (Alt, Gabryszak, and Hennig 2020)). We also explore applying Mask Matching to several *paired-input tasks*, including natural language inference (QNLI (Rajpurkar et al. 2016) and SNLI [2] (Bowman et al. 2015)), paraphrase (PIT2015 (Xu, Callison-Burch, and Dolan 2015) and QQP [3]), word in context (WiC (Pilehvar and Camacho-Collados 2019)), and stance detection (VAST (Allaway and McKeown 2020)). For a fair comparison, all the datasets and the data split are the same as in previous works.

4.2 Comparison Methods

In this research, we compare our method with the following approaches:

State-of-the-art method on the single dataset. We report the previous best results among all the datasets, and these results are directly cited from public papers. It is important to note that our comparisons primarily focus on models that utilize widely used pre-trained language models like BERT, RoBERTa, and LUKE. However, some state-of-the-art methods that rely on pre-training with domain-specific datasets are not directly comparable to our approach.

Fine-tuning PLMs on each dataset. This is the classical solution for NLU tasks. We use the classification-based method for *single-input* tasks, and the semantic matching method for *paired-input* tasks.

Prompt-tuning utilizes a trainable virtual embedding for each label in a given task while keeping other settings the same as Fine-tuning. The primary reason for using a trainable virtual label embedding instead of manually selecting label words is that it requires no manual effort, making it more generalizable to various tasks. Additionally, in many tasks that have a large number of predefined classes, it is often challenging to choose a single discriminating word for each label, particularly in entity typing and relation classification (§1).

Semantic Matching has similar training and testing procedures with Mask Matching. The only difference is that Semantic Matching uses the *max-pooling* over label name as the label representation. [4] The label names are the same as we used in **Mask Matching**.

4.3 Experimental Setup

We use Pytorch (Paszke et al. 2019) and Tesla T4 GPU in our experiments. To ensure the simplicity of our framework, we maintain consistent hyper-parameters across all experiments and observe that results from Mask Matching are consistent across various settings. Specifically, we implement a batch size of 8, with a gradient accumulation of 4, and employ

[1]https://www.cs.umb.edu/ smimarog/textmining/datasets/
[2]https://nlp.stanford.edu/projects/snli/
[3]https://www.quora.com/q/quoradata/

[4]We also explore using the *[CLS]* token in the input side or the *average-pooling* as alternates, but we found that *max-pooling* is the best choice.

Task	Metric	Dataset	#C
Topic Classification	*Accuracy*	R8	8
		R52	52
Semantic Analysis	*Accuracy*	MR	2
		IMDb	2
Entity Typing	*Loose Micro-F1*	FEW-NERD	66
		BBN	47
Relation Classification	*Micro-F1*	TACRED	42
		TAC-REV	42
Natural Language Inference	*Accuracy*	QNLI	2
		SNLI	3
Paraphrase	*Micro-F1*	PIT2015	2
		QQP	2
Word in Context	*Accuracy*	WiC	2
Stance Detection	*Macro-F1*	VAST	3

Table 1: Here is a summary of the datasets we evaluated in our research, with #C representing the number of classes in each dataset. We tested our models on 8 different tasks across 14 datasets, and our metric choices align with those used in previous studies.

the AdamW optimizer (Loshchilov and Hutter 2019), with a learning rate of 1e-5 and a warm-up ratio of 0.2 (Goyal et al. 2017) for all datasets. We use RoBERTa-large [5] in all tasks, except for the entity typing task, where previous findings demonstrated that BERT-large [6] offered superior performance. The training epoch is set to 20, and the maximum input length is limited to 500. While we consider several alternative templates for the *input-prompt* and *label-prompt* in addition to the default template described in §3.2, we observe that the performance differences were negligible (§5.3). To reduce variability, we run each model three times under full training setting and five times in the low-resource scenario. We utilize identical experimental settings for Fine-tuning, Prompt-tuning, Semantic Matching, and Mask Matching.

5 Results

5.1 Full Training Setting

The results of the full training setting on 8 NLU tasks across 14 datasets are presented in Table 1, with the improvements of Mask Matching over Fine-tuning shown in parentheses. Our key observations are listed below:

First, the effectiveness of Mask Matching in comparison to Fine-tuning and Prompt-tuning is remarkable. The performance improvement ranges from 0.2% to almost 2%. For datasets that include label names containing rich semantic information, the improvements are more significant. For example, Mask Matching improves the F1 score from 83.2% to 84.4% on TAC-REV, and from 80.0% to 81.2% on BBN. But the improvements on R8, IMDb, QQP, and natural language inference tasks are relatively small. We attribute this to the following reasons: 1) Strong baseline performances have limited room for improvement; 2) Current design for Mask Matching does not cope well with sentence-paired tasks.[7] Besides, Mask Matching also outperforms Prompt-tuning significantly by fully utilizing the label semantic information. The above results show the effectiveness of Mask Matching, and Mask Matching could serve as a strong baseline for a wide range of NLU tasks.

Second, our intention is not to establish a new state-of-the-art; nevertheless, we report the current SOTA for individual datasets to exhibit the difference between our method and the current leading models. As far as we know, most of these SOTA methods involve task-specific components or training strategies. However, Mask Matching still achieves competitive performances and even obtains slightly better results than comparable SOTA in several tasks, such as topic classification and relation classification. We find that these tasks have a relatively large number of labels, and their labels contain rich information, which is consistent with our conclusion in the previous analysis.

Finally, Semantic Matching is another critical baseline with similar training and testing procedures to Mask Matching. Our experimental results demonstrate that Semantic Matching provides a weaker and less accurate label representation than Mask Matching, despite exploiting the same label semantic information. In particular, Semantic Matching performs poorly on relation classification and entity typing tasks. We attribute this to the impact of word frequency on the representation (Jiang et al. 2022a; Zhou et al. 2022; Zhao, Ma, and Lei 2022; Ding et al. 2021a), indicating that using *max-pooling* or *average-pooling* over label names directly is a sub-optimal approach. We demonstrate that using *input-prompt*, Mask Matching outperforms Semantic Matching, indicating our method's better ability to extract semantic information from label names.

5.2 Low-resource Setting

We also want to explore whether Mask Matching still performs well when the training data is insufficient. The results are shown in Table 3, where we only report the results of Prompt-tuning and Mask Matching since Fine-tuning and Semantic Matching can not achieve desirable performances in most low-resource scenarios. From Table 3 we can see that Mask Matching outperforms Prompt-tuning on 12 of 14 datasets, and the improvements are significant in most cases, especially in entity typing and relation classification. Label names in the above tasks comprise rich semantic information; therefore, with the aid of the *label-prompt*, Mask Matching can take advantage of that information and achieve better results. Improvements in original training sizes that are extensive, such as QNLI, SNLI, and QQP, are considerably minor. In Conclusion, Mask Matching is indeed useful in cases where the label number is significant and label names

[5]https://huggingface.co/roberta-large

[6]https://huggingface.co/bert-lagre-uncased

[7]In fact, this is a common problem faced by prompt-tuning-based methods (Brown et al. 2020; Liu et al. 2021d; Schick and Schütze 2021; Tabasi, Rezaee, and Pilehvar 2022).

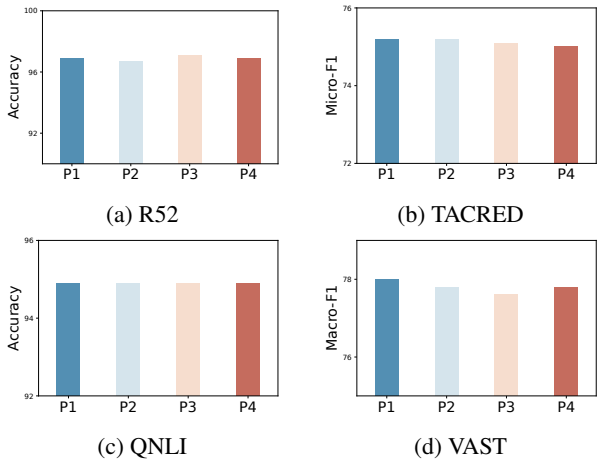
Task	Dataset	Comparable SOTA	Fine-tuning	Prompt-tuning	Semantic Matching	Mask Matching
Topic Classification	R8	98.2 (Lin et al. 2021)	98.1	98.0	98.0	**98.3(+0.2)**
	R52	96.6 (Lin et al. 2021)	96.4	96.7	96.5	**96.9(+0.5)**
Sentiment Analysis	MR	92.5 (Wang et al. 2021)	91.9	91.9	92.0	92.3(+0.4)
	IMDb	97.1 (Ding et al. 2021c)	96.4	96.4	96.4	96.6(+0.2)
Relation Classification	TACRED	75.6 (Li et al. 2023)	74.4	74.3	73.2	75.2(+0.8)
	TAC-REV	84.1 (Li et al. 2023)	83.2	83.3	82.1	**84.4(+1.2)**
Entity Typing	Few-NERD	85.7 (Ding et al. 2021a)	84.6	84.8	84.4	85.2(+0.6)
	BBN	82.2 (Huang, Meng, and Han 2022)	80.3	80.0	79.4	81.2(+0.9)
Natural Language Inference	QNLI(dev)	96.5 (Bajaj et al. 2022)	94.6	94.5	94.2	94.9(+0.3)
	SNLI(dev)	93.1 (Wang et al. 2021)	92.0	92.0	91.7	92.3(+0.3)
Paraphrase	PIT2015	-	83.6	83.7	83.7	**84.1(+0.5)**
	QQP(dev)	93.2 (Bajaj et al. 2022)	92.2	92.1	91.8	92.4(+0.2)
Word in Context	WiC(dev)	71.1 (Liu et al. 2021b)	67.6	68.3	66.1	69.3(+1.7)
Stance Detection	VAST	-	76.8	77.3	77.1	**78.0(+1.2)**

Table 2: Performance of different methods on 14 NLU datasets under the full training setting. We re-implemented some of the methods and cited comparable SOTA results from public papers. The best performances achieved by Mask Matching are denoted in bold. Since previous methods in PIT2015 and VAST datasets used specific PLMs such as SBERT (Reimers and Gurevych 2019), we left the comparable SOTA results blank. To ensure a fair comparison, we report the performance on the development sets of WiC, QNLI, SNLI, and QQP, as previous works (Liu et al. 2019; He, Gao, and Chen 2021; Wang et al. 2021; Bajaj et al. 2022) only presented the single model's performances on the development set. The numbers in parentheses indicate the improvement in performance compared with Fine-tuning. We averaged the results of our methods over three random seeds, and the results we reported are statistically significant with $p < 0.05$.

contain rich semantic information, resulting in a substantial improvement in performance.

5.3 The Sensitivity to Different Label-Prompts

We also investigate the effectiveness of different templates used in the *label-prompt*. The default template is ***P1*** *= [Label Name] is [MASK]*, as we discussed in §3.2. We explore using three additional templates: ***P2*** *= The meaning of [Label Name] is [MASK]*, ***P3*** *= Label Name] means [MASK]*, ***P4*** *= [Label Name] is similar to [MASK]* for comparison purposes. We conduct experiments on both *single-input* and *paired-input* datasets, and the results can be seen in Figure 3. The outcomes indicate that all prompt templates yield similar performances, indicating that different prompt templates have little effect on the results. Furthermore, Mask Matching is immune to various templates in the *label-prompt*. Given that **P1** delivers more stable results than the others, we select this as the default prompt template.



(a) R52 (b) TACRED

(c) QNLI (d) VAST

Figure 3: The sensitivity to different label-prompts. Mask Matching is insensitive to the template used.

5.4 Effects of Enriching Label Names

Previous results in §5.1 and §5.2 suggest that Mask Matching is efficient in leveraging the semantic information contained in label names. Motivated by Li et al. (2022), we explore the possibility of enhancing our approach by adding more related information to label names. To test this, we use FEW-NERD and TACRED datasets, which belong to information extraction tasks and have label names containing important information for prediction. We manually selected two related words for each label as augmentation information, and combined them with the original label name to form the augmented label. Results displayed in Table 4 indicate that incorporating additional information into label names contributes to performance improvement, further demonstrating the utility of label names in the Mask Matching model.

Dataset	Prompt-tuning	Mask Matching
R8	97.3	97.4(+0.1)
R52	92.5	93.0(+0.5)
MR	95.7	95.9(+0.2)
IMDb	90.8	91.4(+0.6)
FEW-NERD	74.7	82.2(+7.5)
BBN	74.3	79.6(+5.3)
TACRED	64.9	67.0(+2.1)
TAC-REV	72.1	73.2(+1.1)
QNLI	91.7	92.1(+0.4)
SNLI	90.2	90.9(+0.7)
PIT2015	75.4	78.3(+2.9)
QQP	88.7	88.7(+0.0)
WiC	58.8	62.4(+3.6)
VAST	75.5	73.8(-1.7)

Table 3: Performance of different methods on 14 NLU datasets under the low-resource setting. We randomly select 10% percent of the whole training set, and keep the development and test sets unchanged. Results are averaged over five random seeds to reduce the randomness.

	FEW-NERD	TACRED
Mask Matching	85.2	75.2
+ Augmentation	**85.6(+0.4)**	**75.4(+0.2)**

Table 4: The experimental results with augmented information on two information extraction tasks. The results show that enriching label names benefits Mask Matching.

6 Discussions

Despite the promising results achieved by Mask Matching, there is still ample opportunity to enhance our method further. To this end, we discuss multiple research directions in this section, with the goal of encouraging readers to consider the broader use of *label-prompt*.

Designing a new framework for paired-input tasks. As we observed in §5.1 and §5.2, Mask Matching does not gain remarkable improvements when dealing with sentence-pair tasks, such as natural language processing and paraphrase. Most prompt-tuning methods (including Mask Matching) directly concatenate two sentences and use a mask token for prediction, which may not be the best choice. One potential solution is to represent two sentences using two mask tokens with joint encoding, and then design a new interactive module or training strategy to get the final prediction via two mask representations.

Automatically extending the label names. Although §5.4 highlights the potential performance improvements resulting from the inclusion of additional relevant information in label names, manually identifying synonyms is time-consuming and challenging without domain knowledge. Automating the extension of label names is, therefore, a promising research direction that could make Mask Matching more robust and generalizable. Furthermore, since the use of Mask Matching requires instances of named labels, extracting important details from input texts and generating appropriate label names automatically is an interesting research area.

Exploring collaborative training using multiple mask tokens. Some works (Wang et al. 2022b; Park et al. 2022) utilize multiple mask tokens for downstream tasks and achieve favorable performances. In this research, we only use one mask token in the *input-prompt* and *label-prompt*. We believe using multiple mask tokens and designing a proper training strategy could achieve better performance.

7 Conclusion

We have presented Mask Matching, a paradigm that matches a mask representation generated from a *input-prompt* with another from a *label-prompt*, to uniformly make predictions for a wide range of natural language understanding tasks. Experimental results on a full training setting show that Mask Matching significantly outperforms its fine-tuning and prompt-tuning counterparts, and obtains on-pair or better results than many state-of-the-art methods. Evaluations in the low-resource scenario and several ablation studies further verify the effectiveness of our method. Our research provides a new perspective to utilize the label semantic information, and we hope Mask Matching could inspire more exploration of the prompting methodology on the label side.

Acknowledgements

This research is supported by the National Key Research And Development Program of China (No. 2021YFC3340101).

References

Allaway, E.; and McKeown, K. 2020. Zero-shot stance detection: A dataset and model using generalized topic representations. *CoRR*.

Alt, C.; Gabryszak, A.; and Hennig, L. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *ACL*.

Bajaj, P.; Xiong, C.; Ke, G.; Liu, X.; He, D.; Tiwary, S.; Liu, T.; Bennett, P.; Song, X.; and Gao, J. 2022. METRO: Efficient Denoising Pretraining of Large Scale Autoencoding Language Models with Model Generated Signals. *CoRR*.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In Màrquez, L.; Callison-Burch, C.; Su, J.; Pighin, D.; and Marton, Y., eds., *EMNLP*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *NeurIPS*.

Chen, X.; Zhang, N.; Xie, X.; Deng, S.; Yao, Y.; Tan, C.; Huang, F.; Si, L.; and Chen, H. 2022. KnowPrompt:

Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction. In Laforest, F.; Troncy, R.; Simperl, E.; Agarwal, D.; Gionis, A.; Herman, I.; and Médini, L., eds., *WWW*.

Dawkins, H. 2021. Marked Attribute Bias in Natural Language Inference. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Findings of ACL/IJCNLP*.

del Arco, F. M. P.; Valdivia, M. T. M.; and Klinger, R. 2022. Natural Language Inference Prompts for Zero-shot Emotion Classification in Text across Corpora. In *COLING*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Ding, N.; Chen, Y.; Han, X.; Xu, G.; Xie, P.; Zheng, H.; Liu, Z.; Li, J.; and Kim, H. 2021a. Prompt-Learning for Fine-Grained Entity Typing. *CoRR*.

Ding, N.; Xu, G.; Chen, Y.; Wang, X.; Han, X.; Xie, P.; Zheng, H.; and Liu, Z. 2021b. Few-NERD: A Few-shot Named Entity Recognition Dataset. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *ACL/IJCNLP*.

Ding, S.; Shang, J.; Wang, S.; Sun, Y.; Tian, H.; Wu, H.; and Wang, H. 2021c. ERNIE-Doc: A Retrospective Long-Document Modeling Transformer. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *ACL/IJCNLP*.

Gao, T.; Fisch, A.; and Chen, D. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *ACL/IJCNLP*.

Goyal, P.; Dollár, P.; Girshick, R. B.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; and He, K. 2017. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *CoRR*.

Han, X.; Zhao, W.; Ding, N.; Liu, Z.; and Sun, M. 2021. PTR: Prompt Tuning with Rules for Text Classification. *CoRR*.

He, P.; Gao, J.; and Chen, W. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *CoRR*.

Huang, J.; Meng, Y.; and Han, J. 2022. Few-Shot Fine-Grained Entity Typing with Automatic Label Interpretation and Instance Generation. In Zhang, A.; and Rangwala, H., eds., *SIGKDD*.

Huang, J. Y.; Li, B.; Xu, J.; and Chen, M. 2022. Unified Semantic Typing with Meaningful Label Inference. In *NAACL*.

Jiang, T.; Huang, S.; Zhang, Z.; Wang, D.; Zhuang, F.; Wei, F.; Huang, H.; Zhang, L.; and Zhang, Q. 2022a. PromptBERT: Improving BERT Sentence Embeddings with Prompts. *CoRR*.

Jiang, Y.; Gao, J.; Shen, H.; and Cheng, X. 2022b. Few-Shot Stance Detection via Target-Aware Prompt Distillation. In *SIGIR*.

Lee, D.; Kadakia, A.; Tan, K.; Agarwal, M.; Feng, X.; Shibuya, T.; Mitani, R.; Sekiya, T.; Pujara, J.; and Ren, X. 2022. Good Examples Make A Faster Learner: Simple Demonstration-based Learning for Low-resource NER. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *ACL*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP*.

Li, B.; Ye, W.; Sheng, Z.; Xie, R.; Xi, X.; and Zhang, S. 2020. Graph Enhanced Dual Attention Network for Document-Level Relation Extraction. In Scott, D.; Bel, N.; and Zong, C., eds., *COLING*.

Li, B.; Ye, W.; Zhang, J.; and Zhang, S. 2023. Reviewing Labels: Label Graph Network with Top-k Prediction Set for Relation Extraction. In Williams, B.; Chen, Y.; and Neville, J., eds., *AAAI*.

Li, B.; Yu, D.; Ye, W.; Zhang, J.; and Zhang, S. 2022. Sequence Generation with Label Augmentation for Relation Extraction. *CoRR*.

Liang, B.; Zhu, Q.; Li, X.; Yang, M.; Gui, L.; He, Y.; and Xu, R. 2022. JointCL: A Joint Contrastive Learning Framework for Zero-Shot Stance Detection. In *ACL*.

Lin, Y.; Meng, Y.; Sun, X.; Han, Q.; Kuang, K.; Li, J.; and Wu, F. 2021. BertGCN: Transductive Text Classification by Combining GCN and BERT. *CoRR*.

Liu, J.; Chen, Y.; and Xu, J. 2022. Low-Resource NER by Data Augmentation With Prompting. In *IJCAI*.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021a. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *CoRR*.

Liu, Q.; Liu, F.; Collier, N.; Korhonen, A.; and Vulic, I. 2021b. MirrorWiC: On Eliciting Word-in-Context Representations from Pretrained Language Models. In Bisazza, A.; and Abend, O., eds., *CoNLL*.

Liu, R.; Lin, Z.; Tan, Y.; and Wang, W. 2021c. Enhancing Zero-shot and Few-shot Stance Detection with Commonsense Knowledge Graph. In *ACL/IJCNLP*.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021d. GPT Understands, Too. *CoRR*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; and Wu, H. 2022. Unified Structure Generation for Universal Information Extraction. In *ACL*.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In Lin, D.; Matsumoto, Y.; and Mihalcea, R., eds., *ACL*.

Nighojkar, A.; and Licato, J. 2021. Improving Paraphrase Detection with the Adversarial Paraphrasing Task. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *ACL/IJCNLP*.

Obeidat, R.; Fern, X. Z.; Shahbazi, H.; and Tadepalli, P. 2019. Description-Based Zero-shot Fine-Grained Entity Typing. In *NAACL-HLT*.

Pang, B.; and Lee, L. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In Knight, K.; Ng, H. T.; and Oflazer, K., eds., *ACL*.

Park, E.; Jeon, D. H.; Kim, S.; Kang, I.; and Na, S. 2022. LM-BFF-MS: Improving Few-Shot Fine-tuning of Language Models based on Multiple Soft Demonstration Memory. In *ACL*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *NeurIPS*.

Pilehvar, M. T.; and Camacho-Collados, J. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In Burstein, J.; Doran, C.; and Solorio, T., eds., *NAACL-HLT*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In Su, J.; Carreras, X.; and Duh, K., eds., *EMNLP*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP*.

Sainz, O.; de Lacalle, O. L.; Labaka, G.; Barrena, A.; and Agirre, E. 2021. Label Verbalization and Entailment for Effective Zero and Few-Shot Relation Extraction. In *EMNLP*.

Schick, T.; and Schütze, H. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *EACL*.

Soares, L. B.; FitzGerald, N.; Ling, J.; and Kwiatkowski, T. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *ACL*.

Tabasi, M.; Rezaee, K.; and Pilehvar, M. T. 2022. Exploiting Language Model Prompts Using Similarity Measures: A Case Study on the Word-in-Context Task. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *ACL*.

Tian, Y.; Chen, G.; Song, Y.; and Wan, X. 2021. Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks. In *ACL/IJCNLP*.

Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019a. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *NeurIPS*.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019b. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *ICLR*. OpenReview.net.

Wang, S.; Fang, H.; Khabsa, M.; Mao, H.; and Ma, H. 2021. Entailment as Few-Shot Learner. *CoRR*.

Wang, Y.; Xu, C.; Sun, Q.; Hu, H.; Tao, C.; Geng, X.; and Jiang, D. 2022a. PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks. In *ACL*.

Wang, Z.; Wang, P.; Liu, T.; Cao, Y.; Sui, Z.; and Wang, H. 2022b. HPT: Hierarchy-aware Prompt Tuning for Hierarchical Text Classification. *CoRR*.

Weischedel, R.; and Brunstein, A. 2005. BBN pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 112.

Wu, X.; Gao, C.; Lin, M.; Zang, L.; and Hu, S. 2022. Text Smoothing: Enhance Various Data Augmentation Methods on Text Classification Tasks. In *ACL*.

Xu, H.; Liu, L.; and Abbasnejad, E. 2022. Progressive Class Semantic Matching for Semi-supervised Text Classification. In *NAACL*.

Xu, W.; Callison-Burch, C.; and Dolan, B. 2015. SemEval-2015 Task 1: Paraphrase and Semantic Similarity in Twitter (PIT). In Cer, D. M.; Jurgens, D.; Nakov, P.; and Zesch, T., eds., *SemEval@NAACL-HLT*.

Zhang, K.; Zhang, K.; Zhang, M.; Zhao, H.; Liu, Q.; Wu, W.; and Chen, E. 2022. Incorporating Dynamic Semantics into Pre-Trained Language Model for Aspect-based Sentiment Analysis. In *Findings of ACL*.

Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; and Manning, C. D. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *EMNLP*.

Zhao, Q.; Ma, S.; and Lei, Y. 2022. Ered: Enhanced Text Representations with Entities and Descriptions. *CoRR*.

Zhong, W.; Gao, Y.; Ding, N.; Liu, Z.; Zhou, M.; Wang, J.; Yin, J.; and Duan, N. 2022. Improving Task Generalization via Unified Schema Prompt. *CoRR*.

Zhou, K.; Ethayarajh, K.; Card, D.; and Jurafsky, D. 2022. Problems with Cosine as a Measure of Embedding Similarity for High Frequency Words. In *ACL*.