ENHANCING CLASS UNDERSTANDING VIA PROMPT-TUNING FOR ZERO-SHOT TEXT CLASSIFICATION

Yuhao Dan[1], *Jie Zhou*[2], *Qin Chen*[*1] , *Qingchun Bai*[3], *Liang He*[1]

[1] School of Computer Science and Technology, East China Normal University, China
[2] School of Computer Science, Fudan Univeirisity, China
[3] Shanghai Open University, China
dan_yh@stu.ecnu.edu.cn, jie_zhou@fudan.edu.cn, {qchen,lhe}@cs.ecnu.edu.cn, qc_bai@foxmail.com

ABSTRACT

Zero-shot text classification (ZSTC) poses a big challenge due to the lack of labeled data for unseen classes during training. Most studies focus on transferring knowledge from seen classes to unseen classes, which have achieved good performance in most cases. Whereas, it is difficult to transfer knowledge when the classes have semantic gaps or low similarities. In this paper, we propose a prompt-based method, which enhances semantic understanding for each class and learns the matching between texts and classes for better ZSTC. Specifically, we first generate discriminative words for class description with **p**rompt **in**serting (`PIN`). Then, a **pro**mpt **m**atching (`POM`) model is learned to determine whether the text can well match the class description. Experiments on three benchmark datasets show the great advantages of our proposed method. In particular, we achieve the state-of-the-art performance on the unseen classes, while maintaining comparable strength with the existing ZSTC approaches regarding to the seen classes.

Index Terms— Zero-shot Text Classification, Prompt Tuning, Semantics Enhancing

1. INTRODUCTION

Text classification has been widely studied due to its fundamental role in natural language processing. Whereas, most of the methods heavily rely on labeled data for training, which are not applicable in real-world scenarios where new classes can emerge without any labeled instances [1]. For example, a large number of topics are bursting in the social media every day, which poses a big challenge to classify the new topics with no labeled instances yet [2].

In recent years, zero-shot text classification (ZSTC) has attracted more attention, which focuses on classifying texts into classes that are unseen during the training stage [3, 4, 5, 6]. Existing methods focus on solving two main issues in ZSTC: (1) how to represent a class with accurate and discriminative descriptions or knowledge; (2) how to effectively learn the matching between texts and class representations on seen/labeled classes, and then extend to unseen/unlabeled classes.

For class representations, the simplest way is to use the label words directly [7, 8], which only contain limited information that can not comprehensively reflect the class semantics. Thus, recent researches focus on incorporating additional knowledge to enhance class representations. Zhang et al. [9] utilized the hierarchical knowledge of class names from ConceptNet [10] to enhance class representations. Puri et al. [8] described the tasks in natural language to facilitate class understanding. Whereas, there are some semantic gaps between the classes and external knowledge or human descriptions, which will affect the performance in the matching stage. Moreover, since the new classes (e.g., topics) may appear suddenly, the knowledge is usually absent from the existing knowledge bases, and the human descriptions can not be prepared in time.

Regarding the matching between texts and classes, recent advanced methods usually utilize the pre-trained language models to encode the concatenation of texts and class representations, and then perform binary classification based on the encoded embedding [11, 12]. Whereas, these methods rely on a large amount of labeled data for learning the model parameters, which is not available in most cases. To alleviate the problem of data deficiency, Yin et al. [11] trained the language model on similar tasks such as natural language inference (NLI), and then fine-tuned the model for text classification. However, severe bias exists in different datasets or domains, which has a side effect on the final performance.

Recently, prompt learning [13] has attracted much attention due to its effectiveness in exploiting pre-trained language models' knowledge to facilitate multiple NLP tasks [14, 15, 16, 17] in low data regime. Inspired by its success, we propose a prompt-based method to enhance class understanding for ZSTC. Specifically, our method consists of two components to resolve the two main issues in ZSTC, namely **p**rompt **in**serting (`PIN`) and **pro**mpt **m**atching (`POM`). The prompt inserting aims to generate class-aware words to enrich class representations with masked language model [18], while prompt matching focuses on devising more efficient templates for matching

[*] Corresponding author.

between texts and class representations. The experimental results indicate the effectiveness of our proposed method, and we achieve a maximum improvement of 6.8% on unseen classes compared with the state-of-the-art methods.

The contributions of our work are as follows: (1) We make full use of the knowledge in the pre-trained language model to represent unseen classes, which obtains class representations with more accurate and richer semantics without relying on any human effort or external knowledge bases. (2) We devise a new effective matching method, which well matches the texts with the corresponding classes with less labeled data during training. (3) Experimental results on three benchmark datasets show the great superiority of our proposed method over the recent advanced methods under the zero-shot scenario.

2. PROPOSED METHOD

In this section, we introduce the details of our proposed method, namely `PINPOM`. The framework is shown in Fig. 1. First, we design a **p**rompt **in**serting (`PIN`) module to generate informative and discriminative words for class description via masked language model (MLM) (Section 2.1). Then, a **p**r**o**mpt **m**atching (`POM`) module is presented to learn the matching between texts and classes (Section 2.2).

For ease of description, we present the notations as well as the task definition as follows. Let S and U denote seen classes (samples can be seen during training) and unseen classes (samples can not be seen during training) respectively. Given training data belonging to S, we aim to train a classifier that can map a text into one of either the seen classes S or the unseen classes U in the inference stage, namely $f(\cdot)$: $X \to Y$, where $Y = S \cup U$.

2.1. Prompt Inserting

Noting that the classifier can only observe the samples from the seen classes during training, how to map the texts to the unseen classes poses a big challenge. Though prompt learning has achieved great success in many few-shot tasks [13], it is still difficult to represent each unseen class in the complete zero-shot scenarios. Since label names only provide limited information, most existing works rely on the human-written class descriptions [19, 11, 12]. However, manual descriptions require expert knowledge and are far from covering enough semantics.

To solve the problem, we present a prompt inserting method (`PIN`), which leverages the knowledge of pre-trained language models to build class descriptions automatically. Specifically, we generate class-aware words as class descriptions with MLM by incorporating the contextual information of class names. More concretely, for each sentence x_y that contains class name y, we feed it into the MLM model (denoted as $\mathcal{M}$) and conpute the word w's probability distribution at the position of the class name y:

$$P(w|x_y) = \mathcal{M}(x_y) \tag{1}$$

It is worth noting that instead of replacing the the class name with the special token "[MASK]" in x_y, we utilize the original text as input, which ensures the generated words semantically consistent with the class name. Take the sentiment classification task as an example, if the label name *love* in the sentence *"I have the feeling of love."* is masked, it would be hard for MLM to predict semantically consistent words for the *love* class, since the text *"I have the feeling of sadness."* is also fluent according to the language model.

With the generated word distribution, we then pick the top-K words and add the probabilities for each instance in the dataset. For the seen classes, we directly use the instances that have the label word in the training data. While for the unseen classes, we search within valid dataset with labels removed, and assume that sentences containing class names belong to the corresponding class. After that, we sort the words in descending order according to the probabilities as the candidate word set W_y^c.

To better ensure the quality of the words for class representations, the words in W_y^c should be discriminative among different classes. In other words, there should be a large enough semantic distance between two similar classes. Whereas, overlapped semantics may exist among W_y^c, as shown in Figure 1, the word "bad" appears in the derived candidate sets of both *anger* and *fear*. To resolve this problem, we refine the candidate word sets by deleting the semantically overlapped words. Specifically, the final candidate words for representing the class y can be obtained as:

$$W_y^c = W_y^c \setminus \bigcup_{y' \in Y, y' \neq y} W_{y'}^c \tag{2}$$

where $\setminus$ and $\cup$ are subtraction and union operations on sets. For each class y, we select the top-k words from y's candidate set as W_y^p, which is regarded as the class description of y.

2.2. Prompt Matching

For ZSTC, we aim to match the input sentence x with names $y \in Y$. Prompt-tuning formulates the problem into a MLM problem. Specifically, We represent class y with class description W_y^p as we mentioned in Section 2.1. We first wrap x and W_y^p into a natural language sentence (prompt) with template $\lambda(\cdot, \cdot)$ as described in Eq. 3.

$$x_p = \lambda\left(x, W_y^p\right) = \text{[CLS]}\ x\ ?\ \text{[MASK]},\ W_y^p\ \text{[SEP]} \tag{3}$$

where "[CLS]" and "[SEP]" indicate the start and end of the template. Then, the MLM gives the probability of each word w in vocabulary appears at the position of "[MASK]" token:

$$P_{\mathcal{M}}(\text{[MASK]} = w|x_p) \tag{4}$$

To map the probability of words to the matching result, we define a map from some of the words to matching labels. Since matched (labeled as 1) or not (labeled as 0) is a binary classification problem, we simply map the words "yes" and

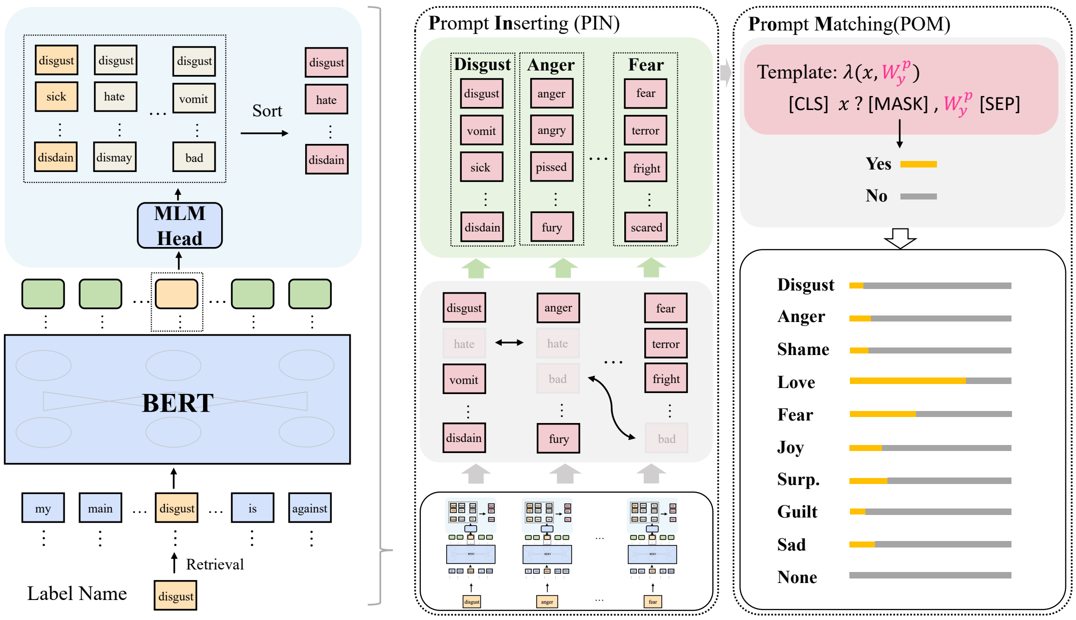

Fig. 1. Illustration of PINPOM. It contains two components: the **Prompt INserting** (PIN) and the **Prompt Matching** (POM). PIN retrieves class-aware words via MLM to enhance class representations by enriching intra-class semantics and removing overlapped inter-class semantics. POM matches input texts with enhanced class representations with prompt learning.

"no" to label 1 and label 0. The probability of x belonging to y is calculated as:

$$P\left(y\,|x\,\right)=\frac{P_{\mathcal{M}}([\text{MASK}]{=}yes|x_p)}{P_{\mathcal{M}}([\text{MASK}]{=}yes|x_p)+P_{\mathcal{M}}([\text{MASK}]{=}no|x_p)}$$

During inference, we calculate $P\left(y\,|x\,\right)$ for each $y \in Y$ given the sentence x, and pick the label with the highest probability as the classification result.

3. EXPERIMENTS

3.1. Experimental Setups

Datasets and Metrics. We conduct the experiments on three publicly available ZSTC datasets released in [11], including Unified Emotion, Yahoo News, and Event-type. We keep the data split strategy and evaluation metrics the same as in [11] so that the derived results are comparable. We report the weighted-f1 score for each dataset on two versions of training data splits for both seen and unseen classes.

Baselines. To comprehensively investigate the effectiveness of our model, we adopt five widely used ZSTC baselines for comparison. We divide the baselines into two parts: classic models and knowledge-enhanced models.

The classic ZSTC models are: (1) Majority: It selects the label of the largest size in the train set as the label of test set, which is the baseline in [11]. (2) Word2Vec: It uses the average of word embeddings as the representations for both the texts and the labels, then determines the text labels with cosine similarity, which is the baseline in [11]. (3) BERT(label): BERT model [20] achieves great success in this task. we input the concatenation of the text and the label name with "[SEP]" into BERT model to train a binary classifier.

The knowledge-enhanced models are: (1) BERT (description): The concatenation of the text and the label's description with "[SEP]" is inputted into BERT model [11]. (2) BERT-MNLI: This model is pre-trained on MNLI dataset [21] and then fine-tuned on ZSTC dataset like BERT (description) [11]. This is the state-of-the-art method for ZSTC.

Parameter Settings. We follow parameter settings in [11]. K and k are set to 5 according to results on valid set.

3.2. Experimental Results

We report the results of baselines and our PINPOM model (Table 1). From this table, we find the following observations. **First**, our PINPOM model outperforms all the classic ZSTC models on the unseen classes by a large margin (see row 1, 2, 3) and achieves comparable results on the seen classes. Moreover, we achieve an average improvement of 4.3% on the whole (both seen and unseen) classes compared with the standard BERT model. The results indicate that our method formulates class descriptions more properly and generalizes well on novel classes. **Second**, from the results compared with knowledge enhanced models (see row 4, 5), the performance of our proposed PINPOM is superior to other methods on the unseen classes. For the seen classes, we can see that BERT+MNLI performs slightly better on a few results. However, BERT+MNLI requires hand-craft class descriptions and post-training on extra MNLI data, which is costly and time-

Table 1. The main results of the baselines and our model. "s" and "u" denote seen and unseen classes respectively. The average score on "s", "u" and "all" (both "s" and "u") is also presented. Best results are marked in bold. Results with * come from the previous paper [11]. Results without * are obtained by averaging over 5 runs with different random number seeds.

	News				Emotion				Event-Type				Average		
	split0		split1		split0		split1		split0		split1				
	s	u	s	u	s	u	s	u	s	u	s	u	s	u	all
Majority*	0.0	10.0	10.0	0.0	0.0	13.3	18.5	0.0	0.0	19.7	0.0	16.4	4.8	9.9	7.3
Word2Vec*	28.1	43.3	43.3	28.1	8.1	5.4	6.2	7.3	10.3	24.7	8.6	23.1	17.4	22.0	19.7
BERT(label)	**73.8**	46.4	**82.6**	29.3	32.2	17.2	**38.1**	17.9	68.6	49.0	**68.7**	49.4	**60.7**	34.9	47.8
BERT(description)*	72.6	44.3	80.6	34.9	**35.6**	17.5	37.1	14.2	72.4	48.4	63.8	42.9	60.4	33.7	47.0
BERT+MNLI*	70.9	52.1	77.3	45.3	33.4	26.6	33.9	21.4	**74.8**	53.4	68.4	33.7	59.8	38.8	49.3
PINPOM	69.6	**54.7**	77.0	**52.1**	32.4	**29.5**	37.6	**23.3**	73.7	**56.6**	66.6	**52.5**	59.5	**44.8**	**52.1**

Table 2. The ablation studies. Best results are marked in bold.

	News				Emotion				Event-type			
	split0		split1		split0		split1		split0		split1	
	seen	unseen	seen	unseen	seen	unseen	seen	unseen	seen	unseen	seen	unseen
PINPOM	69.6	**54.7**	77.0	**52.1**	**32.4**	**29.5**	**37.6**	**23.3**	**73.7**	**56.6**	**66.6**	**52.5**
-PIN	72.3	49.9	**79.9**	38.1	31.6	19.8	37.0	18.7	68.9	52.6	66.1	49.8
-POM	**73.9**	46.8	79.0	44.1	31.2	27.1	37.1	21.4	68.9	52.8	65.4	50.5

consuming. Instead, our PINPOM achieves comparable results on the seen classes without using extra external data, which demonstrates the effectiveness of our method. **Third**, as the class descriptions contain more and more information (from row 3 to row 6), F1 scores gradually increase on the unseen classes and decrease on the seen classes. This may result from that the model knows nothing about the unseen classes, so additional class semantics help the model understand the unseen classes. However, additional semantics may become noise and hinder the model learning the seen classes from training data.

3.3. Further Analysis

Ablation Study. To verify the effectiveness of the components in our model, we do the ablation studies (Table 2). Particularly, we remove the PIN (− PIN) and POM (− POM) from our model, respectively. We observe that: (1) By removing the PIN, the model obtains worse results in most cases, which verifies that the generated class descriptions can enrich the class representation effectively. (2) By removing our POM module, performance on the unseen classes is reduced significantly, which demonstrates that our prompt-based fine-tuning performs better than the classification head in BERT. (3) By using PIN and POM together, our method achieves the best result, which again validates the effectiveness of our method.

Case Study. To better understand the class descriptions generated by PIN, we select several classes and list the words in the descriptions (Table 3). We can see that PIN extracts semantically consistent words with label names by finding either the synonyms (row 1) or special cases (row 3). From inference results, we also find that our PINPOM is able to distinguish subtle differences between semantically similar classes. For the text snippet "Good morning to you, however, it's night time for me, so I am off to bed *hug* Have a great day", PINPOM can correctly classify it to "love", while BERT(label) mistaken it as a semantically similar label "joy".

Table 3. Words generated by PIN

Label	class-consistent Words
joy	fun, joy, happiness, delight, pleasure, excitement
sports	sporting, soccer, athletics, hockey, athletes
infra.	highways, lanes, routes, bridges, pathways

4. CONCLUSION AND FUTURE WORK

In this paper, a novel method for ZSTC task is proposed. Our method contains a prompt inserting module to generate semantically consistent words as class descriptions, facilitating class understanding. Moreover, a prompt matching module is proposed, which effectively matches texts and class descriptions. Extensive experiments on three zero-shot benchmarks show that our method can effectively improve the model's performance. In the future, the following topics will be explored: 1) the selection of the most helpful data for ZSTC from the train set. 2) the effective ways to obtain pseudo labels for data belong to the unseen classes.

Acknowledgement. This research is funded by the National Key Research and Development Program of China (No. 2021ZD0114002), the National Nature Science Foundation of China (No. 61906045), and the Science and Technology Commission of Shanghai Municipality Grant (No. 21511100402).

5. REFERENCES

[1] Bernardino Romera-Paredes and Philip Torr, "An embarrassingly simple approach to zero-shot learning," in *International conference on machine learning*. PMLR, 2015, pp. 2152–2161.

[2] Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary, "Twitter trending topic classification," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 251–258.

[3] Anthony Rios and Ramakanth Kavuluru, "Few-shot and zero-shot multi-label learning for structured label spaces," in *Proceedings of EMNLP*. NIH Public Access, 2018, vol. 2018, p. 3132.

[4] Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert YS Lam, "Reconstructing capsule networks for zero-shot intent classification," in *Proceedings of EMNLP-IJCNLP*, 2019, pp. 4801–4811.

[5] Congzheng Song, Shanghang Zhang, Najmeh Sadoughi, Pengtao Xie, and Eric P Xing, "Generalized zero-shot text classification for icd coding.," in *IJCAI*, 2020, pp. 4018–4024.

[6] Tengfei Liu, Yongli Hu, Junbin Gao, Yanfeng Sun, and Baocai Yin, "Zero-shot text classification with semantically extended graph convolutional network," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 8352–8359.

[7] Pushpankar Kumar Pushp and Muktabh Mayank Srivastava, "Train once, test anywhere: Zero-shot learning for text classification," *arXiv preprint arXiv:1712.05972*, 2017.

[8] Raul Puri and Bryan Catanzaro, "Zero-shot text classification with generative language models," *arXiv preprint arXiv:1912.10165*, 2019.

[9] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo, "Integrating semantic knowledge to tackle zero-shot text classification," in *Proceedings of NAACL*, 2019, pp. 1031–1040.

[10] R. Speer and C. Havasi, "Conceptnet 5: A large semantic network for relational knowledge," *Springer Berlin Heidelberg*, 2013.

[11] Wenpeng Yin, Jamaal Hay, and Dan Roth, "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach," in *Proceedings of EMNLP-IJCNLP*, 2019, pp. 3905–3914.

[12] Z. Ye, Y. Geng, J. Chen, J. Chen, and H. Chen, "Zero-shot text classification via reinforced self-training," in *Proceedings of ACL*, 2020.

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

[14] Allyson Ettinger, "What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 34–48, 2020.

[15] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig, "How can we know what language models know?," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 2020.

[16] Xiang Chen, Xin Xie, Ningyu Zhang, Jiahuan Yan, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen, "Adaprompt: Adaptive prompt-based finetuning for relation extraction," *arXiv preprint arXiv:2104.07650*, 2021.

[17] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harri Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, et al., "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] Abiola Obamuyide and Andreas Vlachos, "Zero-shot relation classification as textual entailment," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 72–78.

[20] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu, "Bert-attack: Adversarial attack against bert using bert," in *Proceedings of EMNLP*, 2020, pp. 6193–6202.

[21] Adina Williams, Nikita Nangia, and Samuel R Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," *arXiv preprint arXiv:1704.05426*, 2017.