Improving Chinese Fact Checking via Prompt Based Learning and Evidence Retrieval

Yu-Yen Ting
Computer Science and Information Engineering
National Central University
Taoyuan, Taiwan
juies309309@gmail.com

Chia-Hui Chang
Computer Science and Information Engineering
National Central University
Taoyuan, Taiwan
chiahui@g.ncu.edu.tw

***Abstract*—Verifying the accuracy of information is a constant task as the prevalence of misinformation on the Web. In this paper, we focus on Chinese fact-checking (CHEF dataset) [1] and improve the performance through prompt-based learning in both evidence retrieval and claim verification. We adopted the Automated Prompt Engineering (APE) technique to generate the template and compared various prompt-based learning training strategies, such as prompt tuning and low-rank adaptation (LoRA) for claim verification. The research results show that prompt-based learning can improve the macro-F1 performance of claim verification by 2%-3% (from 77.62 to 80.29) using golden evidences and 110M BERT based model. For evidence retrieval, we use both the supervised SentenceBERT [2] and unsupervised PromptBERT [3] models to improve evidence retrieval performance. Experimental results show that the micro-F1 performance of evidence retrieval is significantly improved from 11.86% to 30.61% and 88.15% by PromptBERT and SentenceBERT, respectively. Finally, the overall fact-checking performance, i.e. the macro-F1 performance of claim verification, can be significantly improved from 61.94% to 80.16% when the semantic ranking-based evidence retrieval is replaced by SentenceBERT.**

***Index Terms*—fact checking, claim verification, supervised evidence retrieval, prompt based learning, sentenceBERT**

I. Introduction

One of the big problems with user-generated content is that misinformation and false claims cannot be controlled across online media. It relies on the citizens' intelligence to verify the authenticity of any claim, which is challenging and time-consuming. Therefore, automated fact checking plays as a crucial role in addressing the issue created by user-generated contents in the era of Web 2.0.

Automated fact checking consists of two steps: evidence retrieval and claim verification [4]. Evidence retrieval aims to extract pertinent sentences from articles that are related to the claims, while claim verification focuses on assessing the truthfulness of these claims. By leveraging these automated processes, we can expedite and improve their determination of the validity of the claims presented to them.

Previous works on fact checking have mainly focused on English or multilingual contexts. However, we have found that the models employed for Chinese fact checking were rather limited: while the claim verification performance was 78.99% F1 when using golden evidences, the verification performance dropped to 63.47% when automatic evidence retrieval was used [1]. Therefore, exploiting new technique to improve Chinese fact-checking are urgently needed.

Prompt-based learning is a new natural language processing approach that has attracted interest since 2021 [5], where the textual input data is modified by concatenating it with a carefully written text of human readable instructions. The instruction, also called prompt, is designed to narrow the gap between pre-train tasks and downstream tasks.

On the other hand, while adapting to new tasks via fine-tuning is successful, adapting pre-trained models to new tasks by fine-tuning the entire model on smaller labeled datasets is computationally expensive. This is where parameter-efficient fine-tuning (PEFT) [6] comes in as an alternative paradigm to prompting. Specifically, we adopted low rank adaptation (LoRA) for model training.

Finally, while prompting and LoRA can improve the performance of claim verification, the overall fact checking performance is dominated by evidence retrieval. Therefore, we also explore new approaches to improve evidence retrieval performance. As most previous researches utilize unsupervised approaches for evidence retrieval, we first adopted PromptBERT [3] to generate a better sentence representation. Next, to make use of the training data, we follow SentenceBERT [2] to train a binary classifier from claim and sentence pairs.

Our contributions are as follows:

ASONAM '23, November 6-9, 2023, Kusadasi, Turkey
© 2023 Association for Computing Machinery.
ACM ISBN 979-8-4007-0409-3/23/11...$15.00
http://dx.doi.org/10.1145/3625007.3627XXX

1) We studied several prompt-based learning methods as well as PEFT for fact-checking The macro-F1 is improved from 77.62% to 80.29% and 80.62% via p-tuning and LoRA, respectively.
2) We exploited PromptBERT and SentenceBERT for evidence retrieval and improve the retrieval performance from 12.66% to 30.61% and 88.15% F1 respectively.
3) Finally, the overall performance of fact-checking is improved from 61.94% to 79.98% via the combination of SentenceBERT and P-Tuning, which also outperforms the model based on the combination of golden evidences and p-tuning (77.62%).

II. Claim Verification

We first investigate prompt-based strategies and parameter efficient fine tuning for claim verification. Formally, given a claim C and evidence E as input for claim verification, $X = E \oplus C$, the goal is to predict the label $Y \in \{0, 1, 2\}$, denoting “Supported”, “Refuted”, and “Not Enough Information,” respectively.

A. Prompt-Based Learning

The simplest approach for prompt-based strategies is fixed prompt, where a template X_{template} is generated either manually or by large language model through automatic prompt engineering (APE). For manual template, the evidence and claim is presented in the form of attribute and value pair, followed by the hard prompt “Is the claim correct (請問宣稱是對的嗎)?”

For APE, an input example and task prompt:“Generating appropriate instructions based on the following input-output pairs” were fed into GPT 3.0 model to generate multiple templates. By replacing the manual prompt with the generated template, we can fine-tune the used language model to predict the masked tokens for each input example following the same input format, and then evaluate the log probability of the corresponding output in training data. The template with the highest score, which is “Evaluate whether the claim and evidence result in support, refute, or unknown (評估宣稱和證據的結果是支持或反对，或者未知)” is selected as the fixed prompt as well as initial prompt for prompt tuning.

For prompt tuning, we can either fine-tune or freeze the language model parameters. For example, we referred to P-Tuning [7] and used 10 soft tokens (pseudo template) where all prompt parameters are randomly initialized, followed by the selected hard template $X_{template}$ as anchor tokens at the input end (see Table I). When the language model is not large, we can fine-tuned not only the 10 prompt parameters but also the language model parameters. When the language model is too large or computation power is limited, we can freeze the language model parameters and update only prompt parameters. However the later approach usually performs as well as the former.

TABLE I
Inputs of Different Methods

Training Strategies	Input (X)
Fixed-Prompt	[CLS] Evidence:E [SEP] Claim:C [SEP] Hard Prompt [MASK] [MASK]
P-Tuning (v1) (Full-Tune)	[CLS] [soft] [soft]...[soft] [SEP] Evidence:E [SEP] Claim:C [SEP] Anchor Tokens [MASK] [MASK]
P-Tuning (v1) (Freeze)	[CLS] C [SEP] E [SEP] [soft] [soft]...[soft] [SEP]
P-Tuning (v2)	[soft] [soft]...[soft] [CLS] C [SEP] E [SEP]
LoRA	[CLS] C [SEP] E [SEP]

Another alternative way to freeze language model is parameter efficient fine tuning. Specifically, we employ Low-Rank Adaptation (LoRA) [8], which utilizes low-rank decomposition to approximate the attention matrix and fine-tune the model only on the low-rank decomposition to reduce the number of trainable parameters for downstream tasks.

B. Claim Verification Experiments

1) Effect of Full-fine Tuning: For claim verification, we use Micro F1 and Macro F1 as the evaluation metric on the CHEF dataset [1], which consists of 10,000 claims and their associated evidence, collected from four Chinese claim verification websites and one news website. These claims were identified by searching for relevant evidence through search engines. The sources of evidence include the Chinese Internet Joint Rumor Refutation Platform, the Taiwan Fact-Checking Center, the Tencent News Fact-Checking Platform, Piyao, MyGoPen, and the China News Network (Table II).

TABLE II
Dataset statistics

Dataset statistics for CHEF			
Split	Train	Dev	Test
SUP	2877	333	333
REF	4349	333	333
NEI	776	333	333
Total	8002	999	999

Our experiments involve two pre-trained models, including BERT and ERNIE 3.0, as shown in Table III. From the experimental results, we found that prompt tuning, i.e. P-Tuning, outperforms fixed prompt and achieves the best performance in both pre-trained language models. Compared with the baseline performance reported by [1],

TABLE III
Effect of prompt-based learning via full-fine tuning

Pre-trained Language Model	**BERT (110M)**		**ERNIE 3.0 (118M)**	
Approach | Metric	Micro F1	Macro F1	Micro F1	Macro F1
Fine-Tune [1]	78.99 ± 0.82	77.62 ± 1.02	71.15 ± 0.30	70.39 ± 0.30
Hard Prompt	80.10 ± 0.28	79.61 ± 0.23	80.94 ± 0.32	80.38 ± 0.32
P-Tuning v1	80.70 ± 0.56	80.29 ± 0.61	81.40 ± 0.76	80.71 ± 0.81

TABLE IV
Effect of Parameter Efficient Fine-Tune

Parameter Efficient Fine-Tuning (BERT-Large 340M)				
	# Parameters	Time	Micro F1	Macro F1
P-Tuning v1 (freeze)	296K (0.09%)	28 min (↓ 30min)	75.94 ± 1.41	75.67 ± 1.41
P-Tuning v2 (Freeze)	497K (0.15%)	27 min (↓ 31min)	79.36 ± 0.40	78.62 ± 0.39
LoRA (Rank=4)	1,248K (0.39%)	68 min (↑ 10min)	81.28 ± 0.54	80.78 ± 0.62

P-Tuning based on the BERT improves the performance by approximately 2-3%, while the fixed prompt parameter fine-tuning also leads to a 2% improvement. Furthermore, for the ERNIE 3.0 model, P-Tuning demonstrates an improvement of nearly 10%, while fixed prompt parameter fine-tuning shows a 9%-10% improvement.

2) Effect of Parameter Efficient Fine-Tuning: In the parameter-efficient fine-tuning method, we used the 340M large-scale language model BERT and compared the performance of P-tuning with LM freezed and LoRA. Despite a slight decrease in performance for P-Tuning (Freeze), which only utilizes 10 prompt parameters, it still achieved an F1 score of 75%. P-Tuning v2 and LoRA, on the other hand, achieved performance exceeding 80%. Additionally, the training time was reduced by approximately 30 minutes, resulting in time savings (Table IV).

To validate the optimal number of soft prompt tokens for parameter efficient fine-tuning, we conducted experiments with varying numbers of soft prompts. We adjusted the quantity of soft prompts, ranging from few to many, to observe their impact on model performance and determine the best results. In the experiments of parameter efficient fine-tuning, we utilized 10 soft prompt tokens for both P-Tuning v1 and P-Tuning v2. Therefore, we examined the performance with different numbers of soft prompt tokens. When the number of soft prompt tokens was 5, the overall performance decreased due to the limited number of parameters available for fine-tuning. While the quantities of 15 and 10 soft prompt tokens were similar, the best performance was achieved with 10 prompt parameters (Table V).

III. Evidence Retrieval

From previous research [1], we found that evidence retrieval performs relatively weakly in overall fact-checking. After integrating evidence retrieval with claim verification, the performance drops significantly to 63% F1. Therefore, improving the effectiveness of evidence retrieval is a crucial problem that needs to be addressed to enhance overall fact-checking performance.

TABLE V
Effect Of Soft Prompt Token Size in P-Tuning

Parameter Efficient Fine-Tune				
	P-Tuning v1 (freeze)		P-Tuning v2 (freeze)	
Soft tokens	Micro F1	Macro F1	Micro F1	Macro F1
5	71.67	70.61	79.08	78.16
10	**76.68**	**76.27**	**80.38**	**79.60**
15	75.48	74.79	79.58	78.88

Given a document $D = [D_1, D_2, ...D_N]$ and a claim $C = [C_1, C_2, \ldots, C_N]$, our goal is to find sentences in the document that can serve as evidence and are relevant to the claim. Therefore, we need to determine the relevance of each sentence to the claim, and we aim to represent this relevance by predicting a binary label $y \in 0, 1$, indicating whether the sentence is relevant or not. Thus, we treat evidence retrieval as a binary classification task. We remove the golden evidences from the document and split the document into multiple sentences based on punctuation marks. These segmented sentences represent non-evidence sentences. This way, we have a set of non-evidence sentences and a set of golden evidences.

A. Evidence Retrieval Model

For unsupervised evidence retrieval, we use the PromptBERT model to improve sentence representations and further enhance the performance of evidence retrieval. PromptBERT, based on contrastive learning, requires constructing positive samples. Therefore, we employ data augmentation by adding templates to the input as positive examples.

During the training phase, we first input a sentence and define two different templates: "It means [MASK]" denoted as t_1, and "The meaning of this sentence is [MASK]" denoted as t_2. We combine the same sentence with t_1 and t_2 separately, resulting in two prompted sentences containing the templates. Next, we input these two prompted sentences as well as the template sentences t_1

and t_2 into the RoBERTa model, obtaining two prompted sentence vectors and two template vectors.

To prevent the templates from affecting the original sentence representation, we employ a Template Denoise method. We subtract the prompted sentence vectors and the template vectors to obtain the sentence representations. The entire model is updated using a contrastive learning loss function [3]. We aim for the prompted sentence representations to be closer to each other and farther from other sentence representations.

During the testing phase, given a claim C and a sentence S, we combine the claim C with the template t_1 and the sentence S with the template t_2, similar to the training phase. We input the template sentences t_1 and t_2 as well as the prompted claim and sentence into the trained PromptBERT model. We subtract the prompted claim vector from the template vector t_1 and the prompted sentence vector from the template vector t_2. Finally, we obtain the claim vector and the sentence vector.

Next, we calculate the cosine similarity between the two vectors and set a threshold. If the similarity score is below the threshold of 0.8, it indicates a low similarity between the claim and the sentence, and we do not consider it as evidence. Conversely, if the similarity score is above the threshold of 0.8, it indicates an 80% similarity between the sentence and the claim, suggesting their relevance. Therefore, we consider this sentence as evidence.

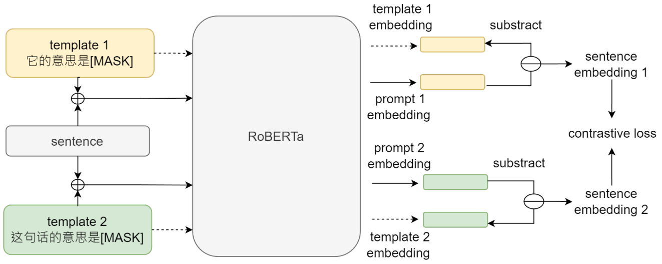

Fig. 1. Architecture of PromptBERT

For supervised evidence retrieval, since we have labels indicating whether each sentence is evidence or not, we use SentenceBERT to learn the relationships between sentences (Fig. 2). During the training phase, given a pair of input claim and sentence, represented as (C, S), we feed them into the pre-trained model RoBERTa. After applying max pooling and obtaining the CLS representation of the sentence, we obtain the claim vector and the sentence vector. Then, we concatenate the claim vector $claim_{emb}$, the sentence vector $sent_{emb}$, and the absolute difference between the two vectors $|claim_{emb} - sent_{emb}|$. This concatenated vector is passed through a Softmax layer to output the predicted class for classification. During the testing phase, we remove the concatenation module and the Softmax layer. Given a pair of claim and sentence (C, S), we pass them through the pre-trained model and apply max pooling to obtain the claim vector and the sentence vector. Finally, we calculate the cosine similarity between these two vectors. We set a threshold of 0.8, and if the similarity score is above 0.8, it indicates relevance between the claim and the sentence, and we consider this sentence as evidence.

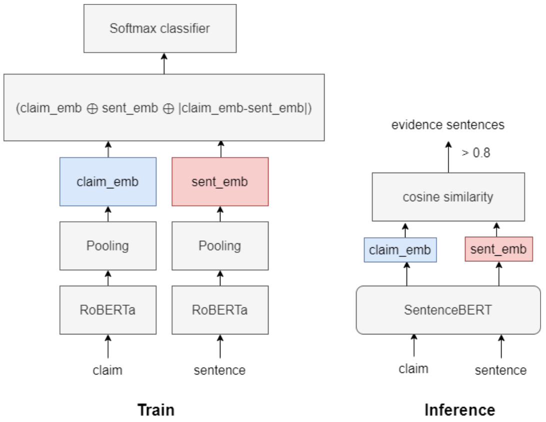

Fig. 2. SBERT Evidence Retrieval

B. Evidence Retrieval Experiment

Since PromptBERT is an unsupervised model, the training data does not contain sentence pairs or labels. During the training process, only individual sentences are used as input. Therefore, our training dataset consists of 8,002 claims and approximately 150,000 sentences, totaling around 160,000 training samples. In contrast, SentenceBERT is a supervised model, and its training data consists of paired claims, sentences, and labels, totaling around 150,000 training samples. Both models will be tested on a set of 999 test samples.

We calculate precision, recall, and micro F1 score between the predicted evidence and the golden evidences for each claim. Then, we calculate the average of precision, recall, and micro F1 score as evaluation metrics. Since both the ground truth and predicted answers may have empty values, we set the precision and recall to 1 if both the ground truth and predicted answers are none. However, if the golden truth is not empty but the model predicts nothing, we set recall to 0. On the other hand, if the golden truth is empty but the predicted answer is not empty, we set recall to 0. We compared the performance of different evidence retrieval methods with the best-performing Semantic Ranker from the CHEF paper (Table VII). The unsupervised PromptBERT model improved sentence representation and further enhanced evidence retrieval performance, achieving an improvement of approximately 18%. The supervised SentenceBERT model also improved evidence retrieval performance and showed a significant improvement of 75%, reaching an F1 score of 88.15%. Both methods effectively improved the precision of evidence retrieval.

TABLE VI
Overall Performance For Fact Verification

Different Evidence on Claim Verification						
	BERT Fine-tune		P-Tuning v1 (full tune)		LoRA (Rank=4)	
	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
Surface Ranker	63.17 ± 1.67	61.47 ± 2.02	66.58 ± 1.51	69.67 ± 1.09	72.95 ± 1.07	70.91 ± 1.37
Hybrid Ranker	63.29 ± 1.65	61.80 ± 2.31	68.02 ± 1.21	70.47 ± 1.00	72.87 ± 0.58	70.88 ± 0.82
Semantic Ranker	63.47 ± 1.71	61.94 ± 1.66	67.16 ± 0.63	69.83 ± 0.63	73.17 ± 1.08	71.14 ± 1.36
PromptBERT (th=0.8)	65.55 ± 1.02	61.58 ± 1.41	63.21 ± 0.51	66.77 ± 0.39	66.91 ± 0.89	63.38 ± 1.32
SentenceBERT(th=0.8)	80.10 ± 1.07	79.72 ± 1.12	80.30 ± 0.24	79.98 ± 0.19	80.54 ± 0.22	80.16 ± 0.16
Golden Evidences	78.99 ± 0.82	77.62 ± 1.02	80.70 ± 0.56	80.29 ± 0.61	81.06 ± 0.28	80.62 ± 0.35

TABLE VII
Evidence Retrieval Performance

Comparison of Different Evidence Retrieval			
	Precision	Recall	Micro F1
Semantic Ranker	10.41	19.22	12.66
PromptBERT	30.73	30.56	30.61
SenteneceBERT	**88.36**	**88.05**	**88.15**

IV. Overall Fact Checking Performance

We integrated evidence retrieval with claim verification and evaluated the overall fact-checking performance, comparing it with fine-tuning approaches. In our study, we chose the best-performing prompt tuning method, P-Tuning, and compared it with traditional BERT fine-tuning methods in terms of performance (Table VI).

We found that the prompt tuning method outperformed traditional BERT fine-tuning methods across different evidence retrieval scenarios. Additionally, SentenceBERT achieved a significant improvement of around 17%-18% in F1 score compared to the performance of Semantic Ranker with BERT fine-tuning, and even further improved by 2% when considering manually labeled golden evidences. This demonstrates that improving evidence retrieval leads to an overall enhancement in fact-checking performance, and even in prompt tuning, the fact-checking performance can surpass that of BERT fine-tuning.

V. Conclusion

In this paper, we focused on enhancing the performance of claim verification and evidence retrieval in Chinese fact-checking. We adopted the APE method to automatically generate suitable templates and employed various prompt learning methods for claim verification, resulting in an improvement from 77.62% to 80.29%, an increase of approximately 2.7% in the macro-F1 score. Furthermore, by adopting LoRA, we were able to achieve a performance improvement to 80.62%. Second, we enhanced the performance of evidence retrieval. With the unsupervised PromptBERT method, we achieved an improvement from 12.66% to 30.61%, successfully boosting the evidence retrieval performance by approximately 18%. Using the supervised SentenceBERT method, we achieved an F1 score of 88.15% in evidence retrieval. Finally, by integrating these two tasks together, we achieved an overall performance improvement from 61.94% to 79.98% for P-Tuning v1, a significant increase of 18%. We even surpassed the performance of manually annotated evidence, improving from 77.62% to 79.98%, with an overall increase of approximately 2.4%.

Acknowledgment

This research is partially supported by National Science and Technology, Taiwan under grant NSC-111-2221-E-008 -078 -MY3.

References

[1] X. Hu, Z. Guo, G. Wu, A. Liu, L. Wen, and P. Yu, "CHEF: A pilot Chinese dataset for evidence-based fact-checking," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Seattle, United States), pp. 3362–3376, Association for Computational Linguistics, July 2022.

[2] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019.

[3] T. Jiang, J. Jiao, S. Huang, Z. Zhang, D. Wang, F. Zhuang, F. Wei, H. Huang, D. Deng, and Q. Zhang, "PromptBERT: Improving BERT sentence embeddings with prompts," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, (Abu Dhabi, United Arab Emirates), pp. 8826–8837, Association for Computational Linguistics, Dec. 2022.

[4] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: a large-scale dataset for fact extraction and VERification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 809–819, Association for Computational Linguistics, June 2018.

[5] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, jan 2023.

[6] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. Raffel, "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning," 2022.

[7] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, "Gpt understands, too," 2021.

[8] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021.