Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Fake news detection via knowledgeable prompt learning

Gongyao Jiang [a], Shuang Liu [b], Yu Zhao [b], Yueheng Sun [b], Meishan Zhang [c,*]

[a] *School of New Media and Communication, Tianjin University, Tianjin 300072, China*
[b] *College of Intelligence and Computing, Tianjin University, Tianjin 300350, China*
[c] *Institute of Computing and Intelligence, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China*

ARTICLE INFO

Keywords:
Fake news detection
Prompt learning
Pretrained language model
Knowledge utilization

ABSTRACT

The spread of fake news has become a significant social problem, drawing great concern for fake news detection (FND). Pretrained language models (PLMs), such as BERT and RoBERTa can benefit this task much, leading to state-of-the-art performance. The common paradigm of utilizing these PLMs is fine-tuning, in which a linear classification layer is built upon the well-initialized PLM network, resulting in an FND mode, and then the full model is tuned on a training corpus. Although great successes have been achieved, this paradigm still involves a significant gap between the language model pretraining and target task fine-tuning processes. Fortunately, prompt learning, a new alternative to PLM exploration, can handle the issue naturally, showing the potential for further performance improvements. To this end, we propose knowledgeable prompt learning (KPL) for this task. First, we apply prompt learning to FND, through designing one sophisticated prompt template and the corresponding verbal words carefully for the task. Second, we incorporate external knowledge into the prompt representation, making the representation more expressive to predict the verbal words. Experimental results on two benchmark datasets demonstrate that prompt learning is better than the baseline fine-tuning PLM utilization for FND and can outperform all previous representative methods. Our final knowledgeable model (i.e, KPL) can provide further improvements. In particular, it achieves an average increase of 3.28% in F1 score under low-resource conditions compared with fine-tuning.

1. Introduction

With the recent development of the Internet, social media platforms, such as Twitter and Facebook, have become popular in daily human life, providing a medium for persons to access and exchange information. Meanwhile, social media drive people to read, post, and propagate news conveniently, also providing an ideal environment for widespread fake news. Fake news that distorts facts and fabricates information leads to negative influences. For instance, various pieces of fake political news will weaken public trust in governments and journalism. Therefore, verifying the authenticity of information is indispensable, which is the goal of fake news detection (FND). Fig. 1 shows two examples of fake news on the dataset PolitiFact (Shu, Mahudeswaran et al., 2020).

FND has attracted great interest for several years. Early works focus on statistical machine learning on handcrafted manual features (Castillo et al., 2011; Kwon et al., 2013; Yang et al., 2012; Zubiaga et al., 2017). In recent years, deep learning has dominated the major advances in the natural language processing (NLP) community. Neural network structures, such as convolutional neural network (CNN), recurrent neural network (RNN), and Transformer, have achieved great successes in feature representation

[*] Corresponding author.
E-mail addresses: jianggongyao@gmail.com (G. Jiang), shuang.liu@tju.edu.cn (S. Liu), zhaoyucs@tju.edu.cn (Y. Zhao), yhs@tju.edu.cn (Y. Sun), mason.zms@gmail.com (M. Zhang).

https://doi.org/10.1016/j.ipm.2022.103029
Received 12 January 2022; Received in revised form 8 July 2022; Accepted 11 July 2022
Available online 26 July 2022

> "Donald Trump plans to step down as president of the united states and resign from office within the next 30 days." Ryan told reporters. "Amid the fury of scandals with Russia... "

> ... FBI uncovers evidence that 62 million Trump voters are all Russian agents anonymous sources within the FBI have revealed to the Times...

Fig. 1. Two examples of political fake news on PolitiFact.

learning (Hochreiter & Schmidhuber, 1997; Kim, 2014; Vaswani et al., 2017). There is a range of studies (Bahad et al., 2019; Dun et al., 2021; Ma et al., 2016; Shu et al., 2019) exploiting deep learning models in representing news content, which is one typical kind of natural language text.

Language model pretraining plays an essential role in the successes of deep learning models, which significantly boosts the performance of various NLP tasks. From the initial word-based pretraining to recent contextualized sentence-level pretraining, several of the text classification tasks have obtained impressive improvement. This improvement should be observed in our FND as well, given that it is also a typical text classification task. Intuitively, FND relies much on external knowledge beyond supervised training instances. As shown in Fig. 1, the two examples are difficult to be recognized as fake without external information. Fortunately, pretrained language models (PLMs), such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), which, to some extent, memorize the vast raw corpus they were pretrained on, can provide a wealth of knowledge for this task. Thus, these PLMs should be recommendable for FND.

The exploration of PLMs has received great interest. From the past studies, there are two main methods: (1) feature-based and (2) fine-tuning. The former extracts features from PLMs, and the latter treats the same models as a startup for further continued training on the target task. For example, ELMo (Peters et al., 2018) is preferable to the feature-based method, while BERT-alike models can achieve better performance by fine-tuning. Currently, BERT-alike models are much better than ELMo; thus, fine-tuning is the mainstream strategy. However, fine-tuning suffers from one major problem that it requires sufficient human-annotated training instances to make it effective. In the realistic scenario, an FND system is strongly correlated with its focused domain. For instance, the model trained on the corpus of gossip or political domain is generally incapable of detecting fake news for the domains such as technology and sports. One suitable way is supervised few-shot learning which trains a model from scratch based on only a small number of annotated instances. The few-shot learning enables to construct FND systems quickly and cheaply for the unexpected domains, reducing the negative influence of fake news.

The performance bottleneck of standard fine-tuning makes PLM-based FND systems difficult to apply in real-world scenarios. Fortunately, prompt learning has been a new paradigm of leveraging PLMs, which keeps the learning process the same as the pretraining objectives, aiming to utilize pretraining information effectively. This paradigm has achieved comparable performance to the standard fine-tuning in various tasks (Han et al., 2021; Liu et al., 2021). In particular, through bridging the gap between pretraining and target task training, prompt learning has achieved remarkable progress in the low-resource setting when only few-shot training instances are available (Brown et al., 2020; Schick & Schütze, 2021a, 2021b). Thus, prompt learning is expected to strengthen the real-scenario FND. Furthermore, knowledge graphs also benefit the FND task via entity representation (Dun et al., 2021). From in Fig. 1, the two examples are both the fake news about Trump's Russia collusion, with different styles and content. A knowledge graph can help extract the key entities such as "Trump" and "Russia", to assist a model to learn from one sample and then generalize to another.

In this work, we propose **K**nowledgeable **P**rompt **L**earning (**KPL**) for FND. On the one hand, the method is highly effective in the low-resource few-shot setting. On the other hand, the model can be competitive when there adequate training instances exist. Unlike the standard fine-tuning that directly outputs the class distributions on the basis of a PLM, prompt learning generates specific answering words related to a cloze question targeted to our task (e.g., "Here is a piece of <_>news".), consistent with the language modeling objective. In addition, we further represent the knowledge graph entities by an n-gram learning manner and incorporate the representations into the prompt learning, to enhance the prompt guidance for detecting fake news.

To simulate the real scenario FND and evaluate our **KPL** approach, we conduct cross-validation and few-shot experiments on two domain-specific datasets: Politifact and GossipCop (Shu, Mahudeswaran et al., 2020; Shu et al., 2017). First, compared with the standard fine-tuning baseline, our prompt learning achieves an average increase of 3.05 in F1 score under few-shot settings and outperforms the baseline with full-scale training corpora. Moreover, the injection of knowledge (i.e., our final **KPL** model) can achieve another F1 score improvement of 2.16 on average under few-shot settings, and the same tendency is observed for full-scale training. Lastly, we compare our model with those of previous studies without pretrained information. **KPL** outperforms all previous methods in low-resource few-shot and full-scale settings, significantly advancing the state-of-the-art results of FND. All our implement codes are publicly available at https://github.com/Zzoay/disinformation_detection for research purposes.

Our main contributions can be summarized as follow:

- In this work, we propose a framework that employs prompt learning to guide FND via effectively utilizing PLM.
- We incorporate knowledge information by entities distilled from one knowledge graph into prompt learning to strengthen the prompt guidance.
- We conduct extensive experiments on two real-world datasets, and experimental results show that our proposed model outperforms state-of-the-art methods under low-resource and data-rich scenarios.

The rest of this article is organized as follows. First, in Section 2, we introduce the related works. Next, Section 3 briefly formulates our targeted task. Then, in Section 4, we present the standard fine-tuning and prompt learning paradigms of exploiting pretrained language models, respectively, and then we present the details of knowledge-enhanced version **KPL**. After that, Section 5 describes our experiments and also offers detailed analyses. In Section 6, we present a discussion of the results and implications. Finally, we make conclusions in Section 7.

2. Related work

2.1. Fake news detection

FND has attracted widespread academic interest (Zhang & Ghorbani, 2020). Following previous works (Meel & Vishwakarma, 2020; Shu et al., 2017), we specify the definition of fake news as false information that spreads under the guise of being authentic news, usually through news outlets or the Internet to gain politically or financially. As easily confused concepts, misinformation and disinformation present differences from fake news. The former is false information due to mistakes or cognitive bias, and the latter is intentionally fabricated information (Meel & Vishwakarma, 2020). Meanwhile, they all consist of text, images, and other carriers. In this paper, our proposed approaches focus on text-based FND but also have the potential to expand to the detection of misinformation and disinformation due to their form similarity.

Early studies of FND focus on designing handcrafted features and then learning patterns by using statistical machine learning methods to distinguish whether a given news is true or false (Zhou & Zafarani, 2020). These works (Castillo et al., 2011; Kwon et al., 2013; Przybyla, 2020; Yang et al., 2012; Zubiaga et al., 2017) leverage textual and social features based on statistical information for detecting fake news. Recently, some works explore the relationship between emotion and news truth. These works focus on engineering emotion and sentiment features for further detecting fake news (Ajao et al., 2019; Zhang et al., 2021). The performance of statistical machine learning methods commonly relies on feature engineering. Nevertheless, these methods can achieve decent performance with few training examples in our experiments, benefiting from their low data demand and feature invariance. Thus, we refer to the text-based feature engineering methods in these works and use statistical machine learning methods as some of our strong baselines.

Deep learning technology has recently been widely explored in various applications, such as NLP and social computing. Benefiting from the strong capabilities of feature extraction and pattern recognition, deep models such as CNN, RNN and Transformer (Kim, 2014; Liu et al., 2016; Vaswani et al., 2017) have been explored intensively in FND (Bahad et al., 2019; Dun et al., 2021; Ma et al., 2016; Samadi et al., 2021; Shu et al., 2019). Among them, Dun et al. (2021) propose a knowledge attention network (KAN) based on Transformer, incorporating the information of a knowledge graph to achieve the recent SOTA performance on benchmark datasets PolitiFact and GossipCop (Shu, Mahudeswaran et al., 2020; Shu et al., 2017). To date, PLM fine-tuning methods have provided simple but strong baselines in FND (Pelrine et al., 2021; Sheng et al., 2021). This paradigm suffers from the gap between the pretraining and tuning processes, showing its performance bottlenecks. Unlike these approaches that do not utilize PLM or utilize it insufficiently, we exploit an effective method (i.e., prompt learning) for utilizing a PLM to guide detecting fake news.

Identifying and curbing fake news at the early stage of its propagation can minimize its pernicious effects (Shu, Zheng et al., 2020). At this condition, FND often faces the problem of labeling data scarcity. Considering this practical need, some studies (Liu & Wu, 2018; Silva et al., 2021) focus on extracting propagation features to detect fake news in its early propagation. For instance, Wang et al. (2021) exploit meta-learning and neural process methods to identify fake news on emergent events with a small set of labeled data. Moreover, several studies (Li et al., 2021a; Shu, Zheng et al., 2020) propose weak social supervision to annotate unlabeled data for training an FND model in conditions of annotated data scarcity. Inspired by but different from these works, we employ prompt learning on a PLM to achieve strong results without leveraging additional social context, image information, or auto-labeled data.

2.2. PLM and utilization approaches

For the past few years, PLMs by self-supervised manners have been the mainstream approach to using large-scale unlabeled data (Qiu et al., 2020). To date, various PLMs, such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019), have been proposed. Previously, there are two main ways to leverage PLMs for target tasks: (1) feature-based and (2) fine-tuning. The former treats the PLM as a feature extractor and fixes it in target task training, and the latter treats the PLM as an initialized backbone for further continued training on downstream tasks. Several results have demonstrated that these tuning approaches, especially fine-tuning, can outperform models trained from scratch for the text classification task (Sun et al., 2019). The FND task is a typical classification task, so the effectiveness of PLM utilization for this task is expected.

A number of studies have explored an alternative tuning paradigm of PLMs, typically called prompt learning. GPT-3 (Brown et al., 2020) demonstrates that given some prompts, a large-scale PLM can achieve decent performance. Afterward, some studies (Schick & Schütze, 2021a, 2021b) reformulate the tuning task to the same as pretraining by converting inputs into handcrafted cloze questions, outperforming the standard fine-tuning method. Considering handcrafted prompts are difficult to find the best choice, later works (Hambardzumyan et al., 2021; Lester et al., 2021; Li & Liang, 2021) focus on utilizing soft prompts, which are randomly initialized and updated during the training process. Given that soft prompts lack interpretability, Han et al. (2021) explore the combination of hard templates and soft tokens. Encouraged by these studies, we exploit prompt learning for utilizing RoBERTa to detect fake news, in which the prompt combines a well-designed prompt template and learnable tokens.

2.3. Knowledge utilization

External knowledge has proven its effectiveness in FND tasks. Several works (Popat et al., 2018; Sheng et al., 2021; Wu et al., 2021) focus on utilizing external evidence to help check the truth of given news. Moreover, some studies, from a consistency perspective (Fung et al., 2021; Sun et al., 2021; Wang et al., 2020), leverage the multimodal data, such as images, to aid in the detection of fake news. Furthermore, some researchers (Cui et al., 2020; Dun et al., 2021; Hu et al., 2021) leverage knowledge graphs to extract essential information from news content to help in FND. Knowledge graphs have excellent flexibility and can be used in situations in which there is no external information such as evidence text and images. For this reason, we use a knowledge graph for text-based FND.

Knowledge graphs consist of a set of interconnected typed entities and have been explored in various fields (Cheng et al., 2020; Li et al., 2021b; Wei et al., 2017; Yang, Huang et al., 2019). They are widely used by means of entity linking (Shen et al., 2014; Zhao et al., 2016). Entity linking with knowledge graph aims to map text mentions to the corresponding entities in a knowledge graph. Applying this entity linking technology, Wang et al. (2018) propose to concatenate knowledge entity embedding and words embedding to enhance the representation of news text. Hu et al. (2021) propose to model the topology of a knowledge graph and align knowledge representation with textual representation. Moreover, Dun et al. (2021) exploit the attention mechanism to align word and knowledge entity representations for FND. Encouraged by but different from these methods, our method integrates knowledge graph information into PLM by a sequence modeling fusion.

Recently, leveraging external knowledge to enhance the performance of PLMs has been extensively studied. Some works focus on the pretraining stage (Liu et al., 2020; Zhang et al., 2019), while some focus on the fine-tuning stage (Guan et al., 2020; Yang, Wang et al., 2019). Furthermore, some studies propose to integrate knowledge information into prompt learning. Hu et al. (2022) integrate knowledge information in verbalizer construction. Chen et al. (2022) inject knowledge entity embeddings into the prompt construction for relation extraction, showing the effectiveness of knowledge integration into the prompt. Inspired by these methods, we incorporate knowledge information into prompt learning (i.e., our KPL method) and explore its application for the FND task.

3. Task formulation

Our research objective is to use the text data of a news article to judge its authenticity. Formally, given a piece of news text $\boldsymbol{x} = [w_1^x, w_2^x, \ldots, w_n^x]$ with n words, the goal of FND is to assign a label $y \in \{0, 1\}$ for the input text, where 0 stands for real news and 1 stands for fake news. One general manner is training a model on a training set $\mathcal{D} = \left\{ \left(\boldsymbol{x}_1, y_1^*\right), \left(\boldsymbol{x}_2, y_2^*\right), \ldots, \left(\boldsymbol{x}_{|\mathcal{D}|}, y_{|\mathcal{D}|}^*\right) \right\}$, where y_i^* denotes the ground-truth label. Meanwhile, annotated data are usually scarce in real-world scenarios, especially at the early stage of fake news propagation. We tend to access only a small subset of training data $\mathcal{D}_{\text{few}} = \left\{ \left(\boldsymbol{x}_1, y_1^*\right), \left(\boldsymbol{x}_2, y_2^*\right), \ldots, \left(\boldsymbol{x}_k, y_k^*\right) \right\}$, where k is the number of training instances and is usually much smaller than the full set size $|\mathcal{D}|$. In this study, we explore an effective method to detect fake news under both data-rich and low-resource conditions.

4. Methodology

Recently, neural models backended with PLMs (Devlin et al., 2019; Liu et al., 2019) have achieved impressive performance in a range of text classification tasks, which should be naturally suited for our task as well. We follow this line of models, adopting typical BERT-based classification models as our baselines, in which the widely exploited fine-tuning and feature-based are applied for modulation. Here, we start from these baselines and then introduce our prompt learning approach to PLM utilization for FND. Lastly, we present its knowledge-enhanced version.

4.1. Fine-tuning and feature-based

Previously, fine-tuning and feature-based are two mainstream ways to utilize PLMs (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020), which have been explored widely. To date, these paradigms have received remarkable success in a number of text classification tasks, and their transfer to FND is natural and straightforward.

Fig. 2(a) illustrates the overview of fine-tuning and feature-based methods for FND. Standard fine-tuning first converts the input text $\boldsymbol{x} = [w_1^x, w_2^x, \ldots, w_n^x]$ into tokens [<cls>, $w_1^x, w_2^x, \ldots, w_n^x$, <sep>]. Then, the PLM is utilized for encoding these tokens to contextualized features:

$$\boldsymbol{h}_{\text{cls}}, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_n, \boldsymbol{h}_{\text{sep}} = \text{PLM-Encoder}\,(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{h}_i$ denotes the hidden state of one token at position i. The special tokens "<cls>" and "<sep>" are reserved inside BERT (Devlin et al., 2019), indicating classification and sentence separation, respectively. The PLM parameters in fine-tuning are updated during the task training, while those in feature-based are fixed.

Following the PLM encoder, a binary classifier is used to compute the probability distribution over the class set Y with a softmax function:

$$p\,(y|\boldsymbol{x}) = \text{softmax}\left(\text{MLP}\left(\boldsymbol{h}_{\text{cls}}\right)\right) \quad (y \in Y) \tag{2}$$

where $\boldsymbol{h}_{\text{cls}}$ is the hidden vector of "<cls>" token, and MLP means the multilayer perceptron module. During the whole training process, the PLM Encoder is initialized using the pretrained parameters, and the fine-tuning approach modulates the entire model by training on the target dataset.

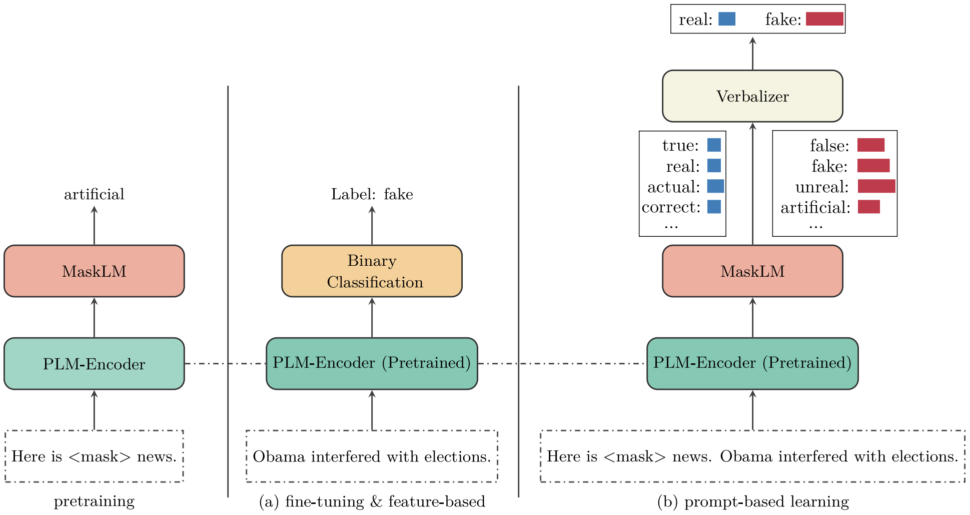


Fig. 2. The general illustrations of pretraining, fine-tuning & feature-based (a) and prompt learning (b).

Table 1
The handcrafted attributes of our prompt learning. We design the prompt template as a summative sentence and set answer words of the real or fake label as 10 relevant words. $Z_{\text{label:real}}$ denotes the set of answer words of real news label, and $Z_{\text{label:fake}}$ denotes the opposite.

Name	Notation	Engineering
Hand-crafted template	$[w_1^{\text{tm}}, \dots, \text{<mask>}, \dots, w_m^{\text{tm}}]$	"Here is a piece of news with <mask> information."
Answer words of real Label	$Z_{\text{label:real}}$	'true', 'real', 'actual', 'substantial', 'authentic', 'genuine', 'factual', 'correct', 'fact', 'truth'
Answer words of fake Label	$Z_{\text{label:fake}}$	'false', 'fake', 'unreal', 'misleading', 'artificial', 'bogus', 'virtual', 'incorrect', 'wrong', 'fault'

4.2. Prompt learning

Previous approaches to PLM utilization, especially fine-tuning, have received great success in data-sufficient conditions, yet they tend to perform poorly in low-resource scenarios (Schick & Schütze, 2021a). One possible reason could be the gap between fine-tuning and pretraining objectives: BERT-alike PLMs are pretrained with a cloze-style objective to learn the distributions of missing words, whereas fine-tuning is to distinguish the target label directly. As a result, the fine-tuning paradigm requires adequate labeled data to tune model parameters for the target task.

To overcome the data-hungry problem of fine-tuning, we suggest another paradigm of PLM utilization, the prompt learning approach. Prompt learning adopts a cloze-style task in the tuning procedure, which is similar to pretraining. Thus, it can leverage the pretraining information more effectively and further enhance the few-shot performance (Schick & Schütze, 2021a). Furthermore, previous works (Hambardzumyan et al., 2021; Han et al., 2021) have demonstrated that a learnable template can enhance the guidance of the prompt. In this subsection, we first introduce the vanilla prompt learning paradigm for FND and then present an extended version with learnable tokens injected into the prompt.

Masked Encoder Unlike fine-tuning, prompt learning converts the training process into the same as the pretraining. In the pretraining process, the input words are randomly masked, allowing the model to recover these masked words. To be consistent with this process, we wrap the input by using a task-related template with one key word being masked. Then, the PLM can calculate the representation of the unknown masked word, which is highly related to our task target. Fig. 2(b) depicts this process in general terms, in which the original input $\boldsymbol{x}$ is concatenated with a slotted template $\boldsymbol{tm}$ to obtain the prompt $\boldsymbol{x}'$:

$$\boldsymbol{x}' = [\boldsymbol{tm}; \boldsymbol{x}] \tag{3}$$

To make the prompt context smooth and instructive, we manually design a summative template (i.e., "Here is a piece of news with <mask>information".), as shown in Table 1. Subsequently, the hidden states of prompt are calculated as:

$$\boldsymbol{h}_1^{\text{tm}}, \dots, \boldsymbol{h}_{\text{mask}}^{\text{tm}}, \dots, \boldsymbol{h}_m^{\text{tm}} | \ \boldsymbol{h}_1^x, \dots, \boldsymbol{h}_n^x = \text{PLM-Encoder}\left(\boldsymbol{x}'\right) \tag{4}$$

where $\boldsymbol{h}_i^{\text{tm}}$ denotes the hidden vector of the ith token in the prompt template, m is the length of the template, and $\boldsymbol{h}_{\text{mask}}$ is the hidden vector of "<mask>" token. $\boldsymbol{h}_j^x$ denotes the hidden vector of input text $\boldsymbol{x}$.

Verbalizer To carry out the procedure of recovering masked words like pretraining, we exploit a module called verbalizer generally, which maps the news label of real and fake into corresponding words. Inspired by Jiang et al. (2020), we conduct a round-trip translation method to paraphrase each label to words with similar meanings, called verbal or answer words typically.

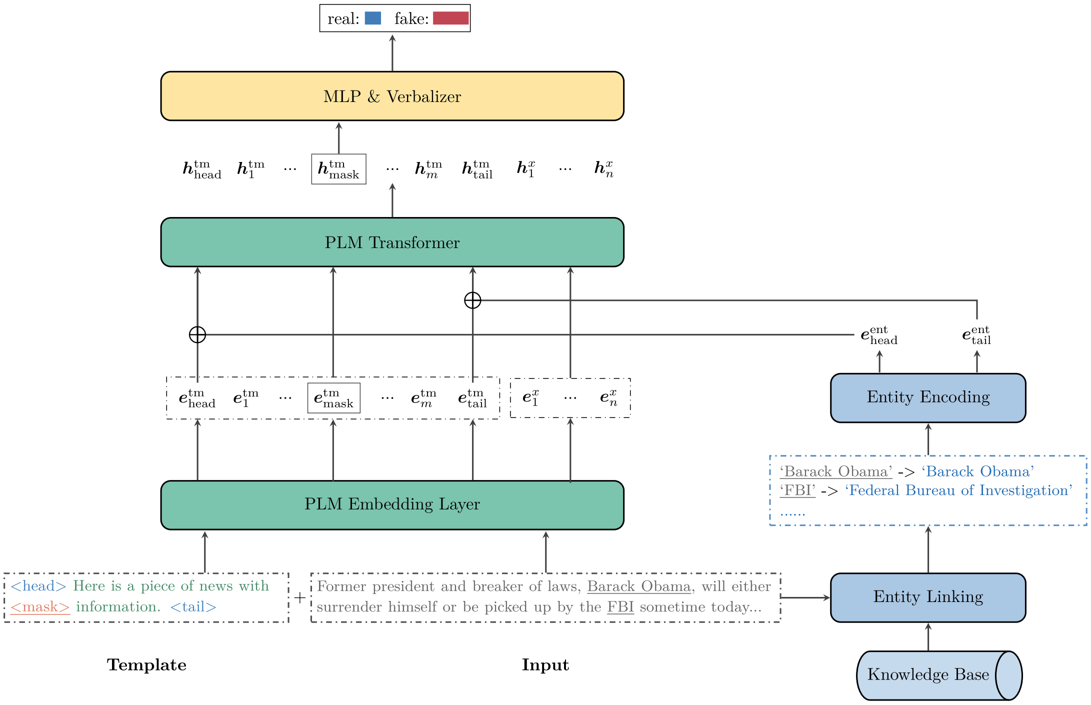

real: 🟦 fake: 🟥
MLP & Verbalizer
h_{head}^{tm} h_1^{tm} ⋯ h_{mask}^{tm} ⋯ h_m^{tm} h_{tail}^{tm} h_1^x ⋯ h_n^x
PLM Transformer
e_{head}^{ent}
e_{tail}^{ent}
Entity Encoding
e_{head}^{tm} e_1^{tm} ⋯ e_{mask}^{tm} ⋯ e_m^{tm} e_{tail}^{tm}
e_1^x ⋯ e_n^x
'Barack Obama' -> 'Barack Obama'
'FBI' -> 'Federal Bureau of Investigation'
......
PLM Embedding Layer
<head> Here is a piece of news with <mask> information. <tail>
+
Former president and breaker of laws, Barack Obama, will either surrender himself or be picked up by the FBI sometime today...
Entity Linking
Knowledge Base
Template
Input

Fig. 3. The overview of our prompt learning with knowledge-aware template representation.

For instance, the label of fake news can be back-translated to "unreal" and "false". We use Google Translation API[1] to purchase back-translated words, and we choose 10 answer words depending on the word frequency provided by the API. Table 1 shows the mapping details between labels and answer words. Then, we predict the distribution of vocabulary on the "<mask>" position and focus on the probability of each answer word z:

$$p\left(z|\boldsymbol{x}'\right) = \mathrm{MLP}\left(\boldsymbol{h}_{\mathrm{mask}}^{\mathrm{tm}}\right) \tag{5}$$

Intuitively, each answer word should have its own weight on the corresponding label. To this end, we assign a learnable weight α_z to each answer word z. Then, the label probability distribution $p(y|\boldsymbol{x}')$ can be calculated using the answer word set Z_y of label y. Eq. (6) formalizes this process.

$$p\left(y|\boldsymbol{x}'\right) = \mathrm{softmax}\left(\sum_{z\in Z_y} \alpha_z p\left(z|\boldsymbol{x}'\right)\right) \tag{6}$$

Learnable Token Injection Several studies (Hambardzumyan et al., 2021; Han et al., 2021) demonstrate that a prompt template with special learnable tokens can make prompt learning more effective. Inspired by that, we further add learnable tokens to our manually designed prompt template.

Concretely, as shown in the lower left corner of Fig. 3, we insert two special learnable tokens "<head>" and "<tail>" into the head and tail of the template:

$$\boldsymbol{tm} = [< \mathrm{head} >, w_1^{\mathrm{tm}}, \ldots, < \mathrm{mask} >, \ldots, w_m^{\mathrm{tm}}, < \mathrm{tail} >] \tag{7}$$

where the embeddings of the two tokens are randomly initialized and updated during the training progress.

Given this template, a new prompt $\boldsymbol{x}'$ can be obtained using Eq. (3). Then, we feed the new prompt $\boldsymbol{x}'$ into the PLM Encoder:

$$\boldsymbol{h}_{\mathrm{head}}^{\mathrm{tm}}, \boldsymbol{h}_1^{\mathrm{tm}}, \ldots, \boldsymbol{h}_{\mathrm{mask}}^{\mathrm{tm}}, \ldots, \boldsymbol{h}_m^{\mathrm{tm}}, \boldsymbol{h}_{\mathrm{tail}}^{\mathrm{tm}} |\ \boldsymbol{h}_1^x, \ldots, \boldsymbol{h}_n^x = \text{PLM-Encoder}\left(\boldsymbol{x}'\right) \tag{8}$$

where $\boldsymbol{h}_{\mathrm{head}}^{\mathrm{tm}}$ and $\boldsymbol{h}_{\mathrm{tail}}^{\mathrm{tm}}$ are the hidden vectors of "<head>" and "<tail>" tokens. Afterward, the contextualized vector $\boldsymbol{h}_{\mathrm{mask}}^{\mathrm{tm}}$ can be integrated into Eqs. (5) and (6) in turn to carry out the same prompt learning procedure above.

[1] https://translate.google.com

4.3. Knowledgeable prompt representation

External knowledge could enhance the performance of PLMs, which has been extensively studied in recent years (Guan et al., 2020; Yang, Wang et al., 2019). Meanwhile, leveraging knowledgeable entities that contain key information can advance the detection of fake news (Dun et al., 2021). To this end, we present a method that incorporates external knowledge representations distilled from a knowledge graph into the prompt template to advance our prompt learning. Fig. 3 reflects our approach exactly.

Knowledge Representation We implement a knowledge embedding process to inject external knowledge into our prompt template. Following the existing work (Dun et al., 2021), we first employ the TagMe (Ferragina & Scaiella, 2010) tool to distinguish the mentions in the input text and link them to corresponding entities in a knowledge database Wikidata (Vrandečić & Krötzsch, 2014).

Next, we exploit a module to represent the extracted knowledge entities that contain one or more words. Motivated by TextCNN (Kim, 2014) that extracts n-gram features via convolutional kernels with different sizes, we concatenate the entities into a sequence $[\boldsymbol{ent}_1, \dots, \boldsymbol{ent}_v]$ and then use a CNN to encode it, aiming to capture both inner-entity and cross-entity information. This process can be formalized as:

$$\boldsymbol{c}_1, \dots, \boldsymbol{c}_u = \mathrm{CNN}\left(\boldsymbol{ent}_1, \dots, \boldsymbol{ent}_v\right) \tag{9}$$

where $\boldsymbol{c}_i$ denotes the n-gram feature compressed using CNN, and u is the length of features after being compressed.

Lastly, to capture bidirectional and long-term information of those inner-entity and cross-entity features, we leverage a bidirectional memory-based recurrent network as the sequential features encoder. In practice, we utilize the bidirectional gated recurrent unit (BiGRU) proposed by Chung et al. (2014), which exploits a gated mechanism to capture long-term information and has lower complexity than the long short-term memory model (Hochreiter & Schmidhuber, 1997). The sequence features are encoded by:

$$\boldsymbol{e}_1^{\mathrm{ent}}, \dots, \boldsymbol{e}_u^{\mathrm{ent}} = \mathrm{BiGRU}\left(\boldsymbol{c}_1, \dots, \boldsymbol{c}_u\right) \tag{10}$$

where $\boldsymbol{e}_i^{\mathrm{ent}}$ is the entity vector with sequence information.

Knowledge Integration We integrate those knowledge features above into our prompt template via a simple representation fusion. To better introduce our knowledge integration in prompt representation, we split the PLM encoder into two parts, Embedding Layer and Transformer module, formalized as:

$$\text{PLM-Encoder}\left(\boldsymbol{x}'\right) = \text{PLM-Transformer}\left(\text{PLM-Embedding}\left(\boldsymbol{x}'\right)\right) \tag{11}$$

Then, we can compute the prompt embeddings by using the PLM Embedding layer:

$$\boldsymbol{e}_{\mathrm{head}}^{\mathrm{tm}}, \dots, \boldsymbol{e}_{\mathrm{mask}}^{\mathrm{tm}}, \dots, \boldsymbol{e}_{\mathrm{tail}}^{\mathrm{tm}} |\ \boldsymbol{e}_1^{x}, \dots, \boldsymbol{e}_n^{x} = \text{PLM-Embedding}\left(\boldsymbol{x}'\right) \tag{12}$$

where $\boldsymbol{e}_i^{\mathrm{tm}}$ denotes the embedding of the ith word in the template. $\boldsymbol{e}_{\mathrm{head}}^{\mathrm{tm}}$, $\boldsymbol{e}_{\mathrm{mask}}^{\mathrm{tm}}$, and $\boldsymbol{e}_{\mathrm{tail}}^{\mathrm{tm}}$ are the embeddings of “<head>” token, “<mask>” token, and “<tail>” token, respectively. $\boldsymbol{e}_j^{x}$ is the jth word vector of input $\boldsymbol{x}$.

In accordance with Eq. (10), the knowledge-aware features $\boldsymbol{e}_1^{\mathrm{ent}}, \dots, \boldsymbol{e}_u^{\mathrm{ent}}$ are output through the BiGRU, which captures both bidirectional and long-term information. Hence, we simply choose the head and tail (indexed as 1 and u) of these sequential features as the representations of knowledge entities. Furthermore, we add these two hidden vectors into the head and tail of the prompt template embeddings:

$$\begin{aligned} \widetilde{\boldsymbol{e}}_{\mathrm{head}} &= \boldsymbol{e}_{\mathrm{head}}^{\mathrm{tm}} + \boldsymbol{e}_{\mathrm{head}}^{\mathrm{ent}} \\ \widetilde{\boldsymbol{e}}_{\mathrm{tail}} &= \boldsymbol{e}_{\mathrm{tail}}^{\mathrm{tm}} + \boldsymbol{e}_{\mathrm{tail}}^{\mathrm{ent}} \end{aligned} \tag{13}$$

Afterward, the knowledge-integrated prompt embeddings are fed into the PLM Transformer:

$$\boldsymbol{h}_{\mathrm{head}}^{\mathrm{tm}}, \dots, \boldsymbol{h}_{\mathrm{mask}}^{\mathrm{tm}}, \dots, \boldsymbol{h}_{\mathrm{tail}}^{\mathrm{tm}} |\ \boldsymbol{h}_1^{x}, \dots \boldsymbol{h}_n^{x} = \text{PLM-Transformer}\left(\widetilde{\boldsymbol{e}}_{\mathrm{head}}, \dots, \boldsymbol{e}_{\mathrm{mask}}^{\mathrm{tm}}, \dots, \widetilde{\boldsymbol{e}}_{\mathrm{tail}}, \boldsymbol{e}_1^{x}, \dots \boldsymbol{e}_n^{x}\right) \tag{14}$$

Following Eq. (5), we feed the contextualized vector $\boldsymbol{h}_{\mathrm{mask}}^{\mathrm{tm}}$ into an MLP module to obtain the answer probability $p(z|\boldsymbol{x}')$. Then, the label probability $p(y|\boldsymbol{x}')$ can be computed via Eq. (6). Finally, we can update the parameters Θ of the entire model during the training process by minimizing the cross-entropy loss function:

$$\mathcal{L} = -\sum \log p\left(y^{*}|\boldsymbol{x}'\right) + \frac{\lambda}{2}\|\Theta\|_2^2 \tag{15}$$

where $\sum$ refers to the sum calculation over all training instances, y^{*} is the ground-truth label, and λ is the coefficient of $L2$ regularizer.

5. Experiment

In this section, we conduct experiments to verify the effectiveness of our approaches. First, we introduce the benchmark datasets and present the implementation details of our experiments. Then, we show and analyze the experimental results of our approach compared with those of the standard fine-tuning and previous methods without PLM utilization. Lastly, we offer detailed analyses for further understanding our proposed methods.

Table 2
Statistics of the news datasets. “#” and “avg.#” denote “the number of” and “the average number of”.

Statistic	GossipCop	PolitiFact
# total news	20956	824
# fake news	4682	379
# real news	16274	445
avg.# words per news	611	1398
avg.# entities per news	115	246

5.1. Data setup

To evaluate the performance of our approach, we conduct experiments on two datasets: PolitiFact and GossipCop. They are included in a benchmark dataset called FakeNewsNet (Shu, Mahudeswaran et al., 2020; Shu et al., 2017) for FND. PolitiFact is a dataset about political news claimed as fake or real by experts. Meanwhile, GossipCop is about entertainment stories with scores on the scale of 0 to 10, and the authors of FakeNewsNet consider the score less than five as fake news. Owing to the copyright restriction, the authors have given the data crawling script. On that basis, we crawl the news contents by utilizing the given script. Then, we leverage the TagMe (Ferragina & Scaiella, 2010) tool to extract knowledge mentions from the texts. The statistic details of the data we obtained are shown in Table 2.

Few-shot settings To simulate low-resource scenarios in the real world, we randomly sample k (2, 4, 8, 16, 100) instances as the training set and build up the development set with the same size, while the remaining samples are utilized as the test set. Considering that different choices of few-shot training set and development set significantly affect the test performance, we repeat this data sampling on 10 random seeds for experiments and use the average score calculated after removing maximum and minimum values as the reported score.

Full-scale settings To reduce distribution bias in the evaluation, we hold out 10% of the news text in each dataset as a development set for tuning the hyperparameters and selecting the best model. For the rest of the dataset, we carry out a 5-fold cross validation and report the average value.

5.2. Implementation details

For model settings, we use RoBERTa$_{\text{base}}$ (Liu et al., 2019), which is an improvement version of BERT, as our PLM. Our implementation of RoBERTa is based on the Hugging Face Transformers Library (Wolf et al., 2020). For the MLP module, we set the hidden size to 200 and the dropout rate to 0.5. In the entity representation module, we set the hidden size to 768. During the training process, the Adam optimizer (Kingma & Ba, 2015) is utilized to optimize the parameters of the model.

In the few-shot settings, the learning rate is 2e−5, and the weight decay is 1e−4. We validate our model every five steps when the shot is 100 and validate every step in other situations. In the full-scale settings, we set a lower learning rate to allow the model to learn more sufficiently on large data, i.e., to 3e−6 empirically. The weight decay is unchanged. For the PolitiFact dataset, we validate our method in the development set after every 100 training steps. For the larger corpus GossipCop, we validate after every 200 training steps. For both few-shot and full-scale settings, we train our model in 20 epochs and choose the checkpoint with the best validation performance to test.

Given that our goal is to detect fake news, we consider fake news as positive examples and use F1-score (F1) to evaluate the classification performance. We also provide the F1 scores of real news in Appendix.

5.3. Comparative methods

As mentioned in Sections 2.1 and 4.1, the application of fine-tuning for text classification can be transferred to FND naturally, expected to produce a remarkable performance as well. Thus, comparing our prompt learning approach to standard fine-tuning (10) for the FND task is indispensable. We also use the feature-based approach (9) as a comparative method, which freezes the PLM parameters and employs the same classifier as that for fine-tuning.

We further compare our method against several strong approaches without PLM utilization, including statistical machine learning methods (1–3) and deep learning methods (4–8). For the statistical machine learning methods (1–3), we design some statistical features (e.g., word numbers, OOV word numbers, entity features, entity repeat count, TF–IDF features, and topic model matrix) on the basis of previous feature engineering experience in the FND task (Castillo et al., 2011; Kwon et al., 2013; Yang et al., 2012; Zhao et al., 2021). For the deep learning methods (4–8), we exploit 300-dim GloVe 6B vectors (Pennington et al., 2014) as the initial word embeddings and entity embeddings.

(1) **DTC** (Castillo et al., 2011): DTC denotes the decision tree classifier, which utilizes the features mined on the news to distinguish the news as fake or real.
(2) **RFC** (Kwon et al., 2013): RFC is the random forest classifier method. Similar to DTC, this method detects fake news based on handcrafted features.
(3) **SVM** (Yang et al., 2012): The support vector machine, based on features extracted from news, tries to use a hyperplane to divide real and fake news in a multidimensional space.

Table 3
Comparison between our approach and previous methods. These scores refer to the F1 scores (%) of fake news. PT denotes our vanilla prompt learning paradigm for tuning the PLM, and KPL is our Knowledgeable Prompt Learning.

Data	Method	Few shot					Full scale
		2	4	8	16	100	
PolitiFact	DTC	39.11	45.69	48.12	53.97	66.49	75.05
	RFC	44.65	45.20	51.59	63.82	72.84	80.03
	SVM	43.25	46.80	42.01	64.18	74.56	82.45
	TextCNN	20.91	19.88	31.91	51.44	75.72	84.34
	BiLSTM	30.36	29.78	45.75	62.11	76.10	85.96
	KCNN	20.89	16.24	31.43	55.71	76.80	85.40
	KLSTM	29.43	20.44	34.96	65.48	76.60	86.41
	KAN	34.05	38.05	47.94	66.27	76.91	87.28
	FB	52.75	58.61	64.97	72.30	80.98	85.76
	FT	50.97	58.32	64.80	71.80	81.23	88.94
	PT	59.61	63.22	65.68	73.39	82.88	89.13
	KPL	**61.25**	**63.92**	**68.34**	**74.60**	**83.51**	**89.52**
GossipCop	DTC	32.25	34.62	32.75	39.45	42.98	50.90
	RFC	30.77	32.13	35.80	38.65	45.52	61.54
	SVM	27.91	30.71	32.19	38.87	45.68	61.85
	TextCNN	23.68	21.22	19.69	16.84	36.26	66.82
	BiLSTM	26.06	30.76	37.58	33.36	41.43	66.78
	KCNN	23.31	19.75	19.12	15.84	38.26	66.51
	KLSTM	26.54	27.76	20.45	20.84	42.25	65.94
	KAN	29.30	31.98	33.56	35.94	43.57	67.35
	FB	36.27	36.71	37.83	40.59	50.90	67.92
	FT	36.75	36.32	37.70	40.38	50.65	68.93
	PT	**38.17**	38.37	39.03	40.69	51.08	68.14
	KPL	37.80	**38.78**	**40.20**	**41.63**	**51.72**	**69.20**

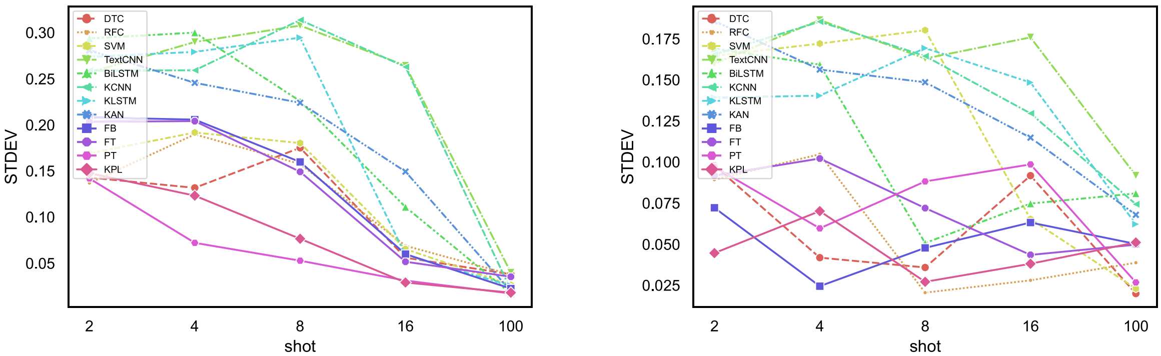
(4) **TextCNN** (Kim, 2014): TextCNN is a widely used deep learning model for text classification, employing convolutional kernels of different sizes to extract text features. We exploit it to judge whether news is real or fake.
(5) **BiLSTM** (Bahad et al., 2019; Hochreiter & Schmidhuber, 1997): The bidirectional long short-term memory model, exploiting memory and gate mechanisms, extracts text features from both directions for each time step.
(6) **KCNN** (Wang et al., 2018): Concatenating the embedding of knowledge entities and words to integrate knowledge information, the KCNN model exploits CNN to learn an integrated representation of given news.
(7) **KLSTM** (Liu et al., 2016; Wang et al., 2018): Inspired by KCNN, we replace its CNN module with BiLSTM, which has achieved better FND performance in previous works (Bahad et al., 2019).
(8) **KAN** (Dun et al., 2021): KAN is the recent SOTA method, integrating knowledge entities and their linked entities into embedding layers. In this study, we utilize 300-dim GloVe vectors as the word and entity embeddings, and we replace the Transformer module with BiLSTM.[2]
(9) **FB** (Peters et al., 2018): FB is the feature-based method of leveraging PLM, and we use RoBERTa as the backbone.
(10) **FT** (Devlin et al., 2019): FT denotes the standard fine-tuning method of PLM utilization based on RoBERTa.
(11) **PT**: PT is our prompt learning method for tuning the PLM. This method contains the answer weights and learnable tokens module mentioned in Section 4.2.

5.4. *Main results*

We carry out comparison experiments under few-shot and full-scale settings. Table 3 shows the main results, from which we can draw the following observations:

Comparison with previous PLM utilization First, we investigate the comparison results between prevalent PLM utilization methods (i.e., fine-tuning and feature-based) and our prompt learning under few-shot settings. According to the results, our prompt learning method (PT) achieves higher F1 scores than the standard fine-tuning (FT) in most settings, in which the average improvement is $\frac{(61.25-50.97)+\cdots}{5\times2}+\frac{(37.80.12-36.75)+\cdots}{5\times2}=3.28$ per point of F1. The improvement becomes more significant when the scale of training data decreases, showing the superiority of prompt learning in low-data scenarios. In addition, the feature-based approach exhibits a slight increase compared with fine-tuning in few-shot settings. Nonetheless, our method can achieve a further improvement compared with it. These experimental results demonstrate the superiority of our method in utilizing PLM information.

[2] In our reproduction works, we find that these settings got a better performance (F1 score in full-scale PolitiFact is improved from 0.85 to 0.87).



(a) PolitiFact (b) GossipCop

Fig. 4. The standard deviation (STDEV) of F1 scores in few-shot settings.

Impact of knowledge integration Second, we examine the effectiveness of knowledge integration on our prompt learning. We observe that our KPL outperforms vanilla prompt learning, which is $\frac{(61.25-59.61)+\cdots}{5\times2}+\frac{(37.80-38.17)+\cdots}{5\times2}=0.96$ per points on average. This result demonstrates that the integrated knowledge can further guide the model for detecting fake news.

Vanilla prompt learning performs better than KPL under the two-shot setting on GossipCop. One possible reason may be that the large gap between the number of training samples and the whole dataset (2/19069), leading to our training knowledge entities having extremely low coverage to generalize over test samples and aggravating the model overfitting with few samples. Nevertheless, the superior performance in most conditions verifies the effectiveness of our approach in the low-data regimes, adapting to early detection of fake news.

Comparison with models trained from scratch Then, we compare our approaches with several representative methods. We can see that both PT and KPL outperform these previous works in all few-shot settings. In particular, KPL achieves a maximum of 16.75% (68.34 – 51.59) improvement in the eight-shot setting on the dataset PolitiFact. The results show the superiority of our methods in detecting fake news.

The deep learning methods without external knowledge and pretraining information perform worse than the statistical methods in low-resource conditions. The possible reason may be that without the help of prior information, the updating procedure of parameters is easily affected by the distribution of small training data, leading to poor model generalization over the test set. The improvement achieved by the knowledge-integrated models, such as KAN, demonstrates that external knowledge can alleviate this problem to certain degrees. However, without pretraining information, these methods show their performance bottlenecks. Particularly, fine-tuning and prompt learning achieve higher F1 scores by PLM utilization. The above facts illustrate the significance of external knowledge and pretraining information in low-resource regimes, and the superior performance of our KPL further proves it.

Full-scale results Finally, we investigate our proposed approach in the full-scale setting. When the training data are sufficient, deep learning methods can outperform statistical methods due to their strong capabilities of linguistic feature extracting. The injection of knowledge graph further improves the performance. Furthermore, the standard fine-tuning on the PLM achieves higher F1 scores than all previous approaches. Notably, our KPL method achieves the best classification performance, which can be attributed to the fact that our approach can effectively utilize both external knowledge and pretraining information.

The aforementioned results demonstrate that our approaches achieve superior performance for FND in both low-resource and data-rich regimes, adapting to real-scenario FND.

5.5. Analysis

In this subsection, we offer detailed experimental analyses for a comprehensive understanding of our proposed **KPL** approach in depth.

5.5.1. Stability of approaches

Owing to the high variance problem in low-resource scenarios, the stability of a model is also an essential metric. In practice, we use the standard deviation (STDEV) of F1 scores to measure the stability of a model. Fig. 4 shows the STDEV of 10 experiments for each model under few-shot settings. We find that the STDEV tends to decrease overall as the training set size increases. Furthermore, we notice that statistical machine learning methods tend to be more stable than deep models learned from scratch. The possible reason could be that deep learning is prone to overfitting in the case of insufficient data, leading to poorer generalization and higher randomness. Particularly, our methods are more stable than fine-tuning, feature-based approach, and learning models from scratch in the vast majority of scenarios. This fact demonstrates the superior stability of our approach.

Table 4
Ablation experimental results. "–AW" means that we remove the learnable answer weights module, and "–LT" means removing learnable tokens. "–DT" refers to that the discrete (hand-crafted) template is removed. "–TM" means the model is directly injected by knowledge entities, without any discrete or learnable template.

Data	Method	Shot					Full scale
		2	4	8	16	100	
PolitiFact	KPL	**61.25**	**63.92**	**68.34**	**74.60**	**83.51**	**89.52**
	–AW	59.54	61.59	68.03	72.41	81.51	89.18
	–LT	57.74	60.14	66.07	70.16	80.17	88.38
	–DT	52.68	54.16	62.90	66.20	77.90	88.32
	–TM	50.17	56.43	62.53	65.37	75.10	87.54
GossipCop	KPL	**37.80**	**38.78**	**40.20**	**41.63**	**51.72**	**69.20**
	–AW	36.89	38.45	39.58	40.77	51.16	68.80
	–LT	36.30	36.53	38.57	39.08	44.73	68.46
	–DT	33.76	33.95	35.49	36.21	40.21	67.58
	–TM	33.49	34.21	34.00	36.37	38.82	66.84

Table 5
Impact of freezing different modules. "All tuned" denotes that all parameters are tuned during the training process, "Prompt frozen" indicates that parameters of the prompt are fixed, and "PLM frozen" refers that we freeze the PLM.

Data	Update method	Shot					Full scale
		2	4	8	16	100	
PolitiFact	All tuned	**61.25**	**63.92**	**68.34**	**74.60**	**83.51**	**89.52**
	Prompt frozen	45.85	49.47	67.35	69.69	83.21	89.10
	PLM frozen	47.14	51.98	59.90	65.61	75.78	73.71
GossipCop	All tuned	**37.80**	**38.78**	**40.20**	41.63	**51.72**	**69.20**
	Prompt frozen	32.48	34.57	39.37	**43.64**	49.43	68.57
	PLM frozen	29.12	32.50	33.69	38.41	47.74	59.43

5.5.2. Ablation study

We conduct an ablation experiment to verify the effectiveness of our proposed components. For *-AW*, we remove the learnable weights of answer words; for *-LT*, we directly remove learnable tokens, only using the manually designed template for prompt learning; for *-DT*, we use only the two learnable tokens, without our hand-designed discrete template; for *-TM*, we carry out prompt learning without any template, with only <mask>token and knowledge entity injection. As shown in Table 4, our method has a performance decay without each module in most settings, showing the importance of the key modules in our method. In addition, we observe that the decay of performance becomes more pronounced in few-shot settings, demonstrating the effectiveness of our proposed modules in low-resource scenarios. Notice that the performance of KPL is comparable to fine-tuning in the absence of the answer weight module. One possible reason is that PLM inherently has a prior preference for masked word prediction, determined by the pretraining process. The answer weight module maps the predicted words to specific labels by weighted summation, to produce more reasonable predictions. Furthermore, the performance decay is more significant in the *-DT* and *-TM* settings, especially in the low-resource scenario. One possible reason is that after losing the guiding role of the template, the model has difficulty understanding the meaning of the predicted <mask>position and therefore generates more incorrect predictions. This result demonstrates the significance of templates in task guidance and knowledge integration.

5.5.3. Impact of modules frozen

Our approach updates the parameters of the PLM and additional prompt embeddings. Some works keep the parameters of prompt fixed (Gao et al., 2021; Schick & Schütze, 2021a), and some studies freeze the parameters of PLM (Lester et al., 2021; Li & Liang, 2021). Hence, we compare our method with these parameter updating methods. Table 5 shows the comparison details.

The experimental results demonstrate that our strategy of updating all parameters performs satisfactorily in most settings. When we freeze the prompt parameters, it works more favorably in some cases, showing the potential of static prompts. Furthermore, the performance decay is more pronounced with PLM frozen than with prompt frozen. This fact suggests that although PLM parameters are intuitively prone to overfitting in low-resource scenarios, freezing them tends to make the model unable to learn from given data.

5.5.4. Variation in knowledge integration

We regard the injected knowledge as entity sequences. Thus, we mainly investigate the impact of using different sequence modeling outputs, such as max pooling, average pooling, and our method, i.e., the output of head and tail. Specifically, if we pool the output of CNN or GRU by average or maximum, then there are four settings in total. We compare these four settings with our method. Extracting the head and tail of the CNN is also a comparative setup. Table 6 shows the comparison results. The average performance of the output through GRU is better than that of CNN due to the sequence modeling capability of GRU. In addition, the method using average pooling is on average better than maximum pooling. One possible reason being that average pooling reduces

Table 6
Impact of different methods for merging knowledgeable sequence representation. “G-” prefix means using the GRU output as knowledge representations, and the “C-” prefix is using CNN. “HT” suffix is directly using the head and tail as knowledgeable tokens. “AVG” means the injected representations are obtained by average pooling, and “MAX” is by maximum pooling.

Data	Merging method	Shot					Full scale
		2	4	8	16	100	
PolitiFact	G-HT	**61.25**	**63.92**	**68.34**	**74.60**	**83.51**	**89.52**
	G-AVG	57.93	61.50	67.18	74.50	82.81	88.65
	G-MAX	55.40	61.22	64.71	71.25	83.15	88.52
	C-HT	58.77	58.95	67.86	70.34	82.37	87.43
	C-AVG	57.75	58.94	67.15	71.89	82.24	87,91
	C-MAX	55.50	58.86	66.85	70.75	82.32	88.09
GossipCop	G-HT	**37.80**	**38.78**	**40.20**	**41.63**	**51.72**	**69.20**
	G-AVG	36.38	37.23	39.47	40.14	47.28	68.43
	G-MAX	36.20	36.53	38.57	39.08	44.73	68.17
	C-HT	36.39	37.20	39.26	40.15	46.29	67.77
	C-AVG	36.44	37.20	39.30	38.67	45.72	67.53
	C-MAX	36.05	37.08	39.24	40.12	46.31	67.27

Table 7
F1 scores under zero-shot and two-shot settings. “Vanilla” denotes leveraging PLM for direct prediction or training with two samples (standard fine-tuning). And “+P” denotes adding a manual design prompt to guide prediction, “+KP” denotes injecting knowledge into the prompt template.

Data	Method	Zero-shot	Two-shot
PoitiFact	Vanilla	25.26	50.97
	+P	36.73	59.61
	+KP	**36.89**	**61.25**
GossipCop	Vanilla	15.53	36.75
	+P	22.97	**38.17**
	+KP	**23.35**	37.80

the loss of information compared with maximum pooling. It is important to note that the best average performance is obtained using the output of the head and tail entities. One reason may be that this approach is consistent with the way we integrate knowledge; i.e., we insert knowledge entities into the head and tail of the template in the form of learnable vectors. Furthermore, the method we applied, i.e., using the head and tail vectors of the GRU output as knowledge features, achieves the best results. This is in line with our analysis above.

5.5.5. Zero-shot performance

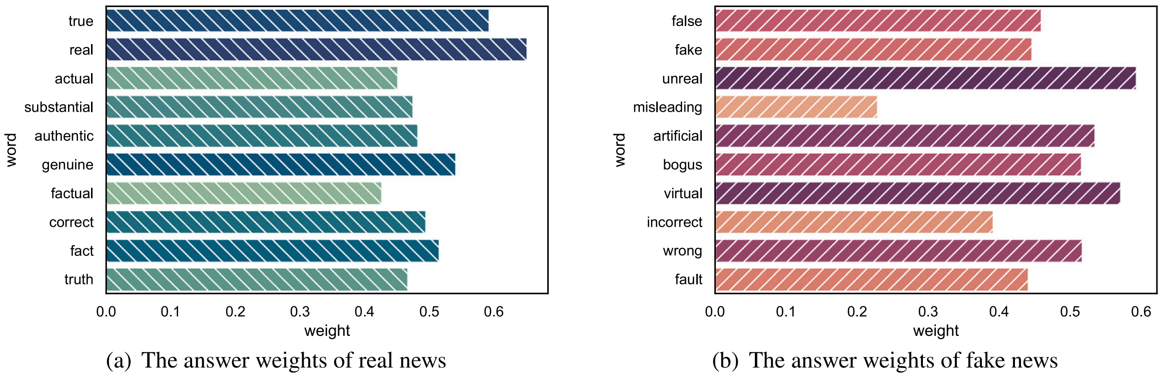
Recently, GPT-3 (Brown et al., 2020) has achieved decent performance in zero-shot settings via the guidance of a given prompt. Motivated by that, we evaluate the zero-shot performance of our methods. We conduct zero-shot experiments on PolitiFact and GossipCop by testing the F1 score on all samples without training. For the knowledge representation module, the random initialization of an entity encoding model has adverse effects in zero-sample scenarios, which is intuitive. Thus, we directly use the sum of entity embeddings as the knowledge representations instead. To reflect the zero-shot performance of prompt learning more intuitively, we also present the two-shot experimental results for comparison. From Table 7, our KPL (+KP) and prompt learning (+P) outperform the PLM direct prediction (Vanilla) in the no-training-data condition, showing the potential of prompt learning in zero-shot FND. Compared with the two-shot setting, the zero-shot performance is significantly reduced, showing that there is room for further improvement of the zero-shot performance of the prompt learning method.

5.5.6. Visualization of answer word weights

We also analyze the weight of each answer word after training. In practice, we average the answer weights under few-shot settings. As shown in Fig. 5, “true” and “real” have the highest weights among the answer words of real news, which is consistent with our intuition. Among the words with the label of fake news, the answer word “unreal” has the highest weight, while “fake” itself has a small weight. One possible reason could be that the PLM has a weak preference for the word “fake”. When we paraphrase it to other words, the model finds a more appropriate word. This result demonstrates that PLMs may have a different perception and understanding of words from humans.

6. Discussion

We proposed KPL for FND, an effective method in both low-resource and data-rich scenarios. Our method can be divided into the following processes: The input text is aligned to a knowledge graph for further knowledgeable representation learning. Then, the knowledge representation is fed into a task-oriented template to build a prompt. Lastly, a PLM encodes the prompt and the output is



(a) The answer weights of real news

(b) The answer weights of fake news

Fig. 5. The weights of answer words.

fed into an MLP module to obtain the word distribution, which is mapped to the specific class of news. In this section, we first analyze the connections and differences between our approach and existing studies. Afterward, we introduce the theoretical implications of our study and how the proposed method can help improve practical applications. Finally, we summarize the limitations and the potential improvements in the future.

6.1. Connection and comparison with existing works

Our proposed KPL can achieve satisfactory performance for FND tasks in different data scenarios. Previous FND methods can be summarized in four categories: (1) a statistical learning approach based on hand-designed features (Castillo et al., 2011; Kwon et al., 2013; Yang et al., 2012; Zubiaga et al., 2017), (2) a deep representation learning approach learned from scratch (Bahad et al., 2019; Kim, 2014), (3) an approach incorporating knowledge graphs (Dun et al., 2021; Wang et al., 2018), and (4) a PLM fine-tuning approach (Devlin et al., 2019; Liu et al., 2019). Our method can be considered a combination of 3 and 4, but the tuning approach is prompt learning, different from the standard fine-tuning. The tuning process of prompt learning is consistent with the pretraining stage, leading to more effective utilization of pretraining information than the standard fine-tuning. Our comparison results between prompt learning and fine-tuning in Section 5.4 validate the above intuition.

Our knowledge incorporating method is based on entity linking (Zhao et al., 2016). It is consistent with KAN (Dun et al., 2021), one of our strong baselines. Differently, we use only the entities of the knowledge graph compared with this method and do not need the neighbor relationship of the entities, which has more efficiency and flexibility while achieving better results. Furthermore, the existing works lack attention to FND in low-resource scenarios, which we believe is a reality that cannot be ignored. To this end, we explore our approach in low-resource scenarios. The performance improvement is pronounced in the few-shot setting, revealing the potential of combining prompt learning and knowledge graphs in low-resource scenarios.

6.2. Contributions to future research

In this paper, we provide a new perspective for the study of FND, i.e., exploring FND from the view of prompt learning. FND is a typical classification task, which is naturally suitable to leverage prompt learning. To our best knowledge, this is the first work that uses prompt learning for FND. Thus, our method can be a new strong baseline approach for subsequent research. Currently, the research on prompt learning is still in the developing stage, which means that after we validate the effectiveness of prompt learning in FND, further advanced prompt learning methods can be explored for this task.

The approach of incorporating knowledge entities into templates is generalizable and has the potential to be extended to other tasks. Our method could be explored in other classification tasks due to its superior performance in FND, a typical classification task. In addition, this study has demonstrated the effectiveness of incorporating knowledge graph information into prompt learning, which provides support for subsequent related studies.

6.3. Implications on system design

Our study provides three views on FND system design: First, a real-world system should cope with low-resource scenarios. The scarcity of annotated data is a real-world scenario problem, which is even more pronounced for systems that need to detect fake news in real time. Second, the prompt learning paradigm is promising for low-resource FND. Our approach is adaptable to low-resource conditions and has the ability to perform FND in real-world situations. Third, an effective FND system should have some knowledge, which can be implicit knowledge obtained from pretrained models or explicit knowledge extracted from knowledge graphs or both of them. The improvement in performance with effective use of the PLM and the addition of knowledge graph information validates this view.

6.4. Limitations and future work

There is still some room for improvement in our work. First, our method allows for investigations in other representative benchmarks and datasets with various categories. In addition, the knowledge incorporation approach presented in this paper is relatively simple and does not explicitly mine knowledge to guide the process of FND. Furthermore, multimodal approaches are becoming more common in FND and can effectively improve FND performance by extracting and fusing various types of features in news. The knowledge graph used in this study can be regarded as a modality beyond news text, and the introduction of other modalities, such as images and social networks, may be a part of future work.

As of now, there are some recent and valuable works that we have not considered, owing to the completion date of this work. For instance, the problem of generalization has been an issue that needs attention in FND, and our work considers the low-resource generalization. Mosallanezhad et al. (2022) use reinforcement learning to address cross-domain generalization in FND. Zhu et al. (2022) investigate the entity bias and propose a debiasing framework for future generalization. In addition, some recent works have also introduced external information. Xu et al. (2022) propose to use graph neural networks to represent the key information in evidence texts. Sheng et al. (2022) consider the news environment and propose news environment perception for FND. In general, the generalization problem and the introduction of external information will be the future research trend of FND.

7. Conclusion

In this paper, we proposed **KPL** for FND. The key idea is to apply prompt learning on the PLM and further enrich prompt template representations with entity knowledge. To the best of our knowledge, this is the first work in exploiting prompt learning for FND. Compared with previous approaches, **KPL** is highly effective in both low-resource and data-rich regimes for twofold. On the one hand, our prompt learning leverages an additional prompt to guide the pretrained model execution. Meanwhile, it adopts one consistent objective with language model pretraining, significantly reducing the gap between pretraining and target task training. On the other hand, the injection of knowledge graph information by entities in the well-designed prompt can further enhance our model to detect fake news.

To evaluate our approach, we conducted experiments on two benchmark datasets. We selected statistical machine learning and deep learning methods for comparison, and several of these models are also incorporated with the same external knowledge. Experimental results showed that the prompt learning method leads to excellent performance compared with previous methods under both low-resource and data-rich settings. In addition, our experiments also demonstrated that external knowledge is highly beneficial for prompt learning in detecting fake news. Finally, we can obtain new state-of-the-art results based on our **KPL** model for FND. Furthermore, we conducted detailed analyses for a comprehensive understanding of our method.

CRediT authorship contribution statement

Gongyao Jiang: Conceptualization, Methodology, Software, Writing – original draft. **Shuang Liu:** Supervision, Writing – review & editing. **Yu Zhao:** Writing – review & editing. **Yueheng Sun:** Funding acquisition, Supervision. **Meishan Zhang:** Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NFSC) grants 62176180 and 61802275.

Appendix. More details of main results

Here we provide the F1 scores of the real class, reported in Table 8. From this table, we can observe that our prompt learning (PT) and knowledgeable prompt learning (KPL) achieve the best performance under most settings. This is in line with Section 5.4 (Main Results). In addition, the provided F1 scores of real and fake classes can be used to calculate the Macro-F1, which is also a common metric for the FND task, to facilitate future research.

Table 8
The F1 scores of real news.

Data	Method	Few shot					Full scale
		2	4	8	16	100	
PolitiFact	DTC	42.39	45.49	56.05	59.55	70.65	78.67
	RFC	40.20	46.46	61.10	60.09	78.00	83.97
	SVM	35.72	46.74	59.76	61.49	78.04	85.33
	TextCNN	44.56	45.78	48.49	46.70	76.28	86.29
	BiLSTM	45.22	45.91	42.33	40.65	76.38	86.99
	KCNN	44.78	46.97	55.19	55.05	79.22	87.25
	KLSTM	44.50	45.34	54.43	48.43	79.20	87.60
	KAN	43.86	44.75	53.57	60.23	79.45	88.45
	FB	43.60	43.61	54.40	63.78	83.28	86.62
	FT	44.33	45.60	58.39	69.40	84.35	90.24
	PT	44.66	45.26	59.54	72.40	85.16	90.80
	KPL	**46.17**	**48.18**	**63.06**	**72.55**	**86.09**	**91.00**
GossipCop	DTC	48.25	50.02	52.13	46.91	71.59	84.14
	RFC	48.97	53.65	51.29	51.59	73.38	88.26
	SVM	44.08	50.69	52.09	55.35	79.00	85.97
	TextCNN	49.70	54.06	54.67	57.57	74.34	88.79
	BiLSTM	49.90	48.19	41.60	52.51	70.64	89.16
	KCNN	52.55	54.42	56.00	57.55	72.93	89.32
	KLSTM	51.78	49.03	55.28	57.55	71.72	89.76
	KAN	50.16	47.89	50.55	53.37	73.86	90.02
	FB	50.14	51.19	54.02	53.27	82.66	87.56
	FT	52.21	53.36	55.64	54.49	82.11	89.76
	PT	52.68	54.78	56.08	56.41	82.69	90.36
	KPL	**54.15**	**56.63**	**56.70**	**58.20**	**84.70**	**90.48**

References

Ajao, O., Bhowmik, D., & Zargari, S. (2019). Sentiment aware fake news detection on online social networks. In *ICASSP 2019 - 2019 IEEE international conference on acoustics, speech and signal processing* (pp. 2507–2511). http://dx.doi.org/10.1109/ICASSP.2019.8683170.

Bahad, P., Saxena, P., & Kamal, R. (2019). Fake news detection using bi-directional LSTM-recurrent neural network. *Procedia Computer Science*, *165*, 74–82. http://dx.doi.org/10.1016/j.procs.2020.01.072, 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION, 2019 November 11-12, 2019.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., 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., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems, Vol. 33* (pp. 1877–1901). Curran Associates, Inc..

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on Twitter. In *WWW '11*, *Proceedings of the 20th international conference on world wide web* (pp. 675–684). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1963405.1963500.

Chen, X., Zhang, N., Xie, X., Deng, S., Yao, Y., Tan, C., Huang, F., Si, L., & Chen, H. (2022). KnowPrompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW '22*, *Proceedings of the ACM web conference 2022* (pp. 2778–2788). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3485447.3511998.

Cheng, L., Wu, D., Bing, L., Zhang, Y., Jie, Z., Lu, W., & Si, L. (2020). Knowledge graph empowered entity description generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing* (pp. 1187–1197).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, abs/1412.3555. arXiv:1412.3555.

Cui, L., Seo, H., Tabar, M., Ma, F., Wang, S., & Lee, D. (2020). Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 492–502).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171–4186).

Dun, Y., Tu, K., Chen, C., Hou, C., & Yuan, X. (2021). KAN: Knowledge-aware attention network for fake news detection. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35* (1), (pp. 81–89).

Ferragina, P., & Scaiella, U. (2010). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 1625–1628).

Fung, Y., Thomas, C., Reddy, R. G., Polisetty, S., Ji, H., Chang, S.-F., McKeown, K., Bansal, M., & Sil, A. (2021). Infosurgeon: Cross-media fine-grained information consistency checking for fake news detection. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 1683–1698).

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 3816–3830). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.acl-long.295.

Guan, J., Huang, F., Zhao, Z., Zhu, X., & Huang, M. (2020). A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, *8*, 93–108. http://dx.doi.org/10.1162/tacl_a_00302.

Hambardzumyan, K., Khachatrian, H., & May, J. (2021). WARP: Word-level adversarial reprogramming. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 4921–4933). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.acl-long.381.

Han, X., Zhao, W., Ding, N., Liu, Z., & Sun, M. (2021). PTR: Prompt tuning with rules for text classification. arXiv preprint arXiv:2105.11259.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, S., Ding, N., Wang, H., Liu, Z., Wang, J., Li, J., Wu, W., & Sun, M. (2022). Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long Papers)* (pp. 2225–2240). Dublin, Ireland: Association for Computational Linguistics.

Hu, L., Yang, T., Zhang, L., Zhong, W., Tang, D., Shi, C., Duan, N., & Zhou, M. (2021). Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 754–763).

Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, *8*, 423–438. http://dx.doi.org/10.1162/tacl_a_00324.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1746–1751). Doha, Qatar: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/D14-1181.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio, & Y. LeCun (Eds.), *3rd international conference on learning representations*. URL http://arxiv.org/abs/1412.6980.

Kwon, S., Cha, M., Jung, K., Chen, W., & Wang, Y. (2013). Prominent features of rumor propagation in online social media. In *2013 IEEE 13th international conference on data mining* (pp. 1103–1108). http://dx.doi.org/10.1109/ICDM.2013.61.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 3045–3059). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Li, Y., Lee, K., Kordzadeh, N., Faber, B., Fiddes, C., Chen, E., & Shu, K. (2021). Multi-source domain adaptation with weak supervision for early fake news detection. In *2021 IEEE international conference on big data* (pp. 668–676). http://dx.doi.org/10.1109/BigData52589.2021.9671592.

Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing* (pp. 4582–4597). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.acl-long.353.

Li, Y., Zou, B., Li, Z., Aw, A. T., Hong, Y., & Zhu, Q. (2021). Winnowing knowledge for multi-choice question answering. In *Findings of the association for computational linguistics: EMNLP 2021* (pp. 1157–1165). Punta Cana, Dominican Republic: Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692. arXiv:1907.11692.

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. In S. Kambhampati (Ed.), *Proceedings of the twenty-fifth international joint conference on artificial intelligence* (pp. 2873–2879). IJCAI/AAAI Press.

Liu, Y., & Wu, Y. B. (2018). Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In S. A. McIlraith, & K. Q. Weinberger (Eds.), *Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence* (pp. 354–361). AAAI Press, URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16826.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., & Tang, J. (2021). GPT understands, too. arXiv preprint arXiv:2103.10385.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., & Wang, P. (2020). K-BERT: enabling language representation with knowledge graph. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence* (pp. 2901–2908). AAAI Press.

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K. F., & Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks. In *IJCAI international joint conference on artificial intelligence, Vol. 2016* (pp. 3818–3824).

Meel, P., & Vishwakarma, D. K. (2020). Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, *153*, Article 112986.

Mosallanezhad, A., Karami, M., Shu, K., Mancenido, M. V., & Liu, H. (2022). Domain adaptive fake news detection via reinforcement learning. In F. Laforest, R. Troncy, E. Simperl, D. Agarwal, A. Gionis, I. Herman, & L. Médini (Eds.), *WWW '22: The ACM web conference 2022* (pp. 3632–3640). ACM, http://dx.doi.org/10.1145/3485447.3512258.

Pelrine, K., Danovitch, J., & Rabbany, R. (2021). The surprising performance of simple baselines for misinformation detection. In J. Leskovec, M. Grobelnik, M. Najork, J. Tang, & L. Zia (Eds.), *WWW '21: the web conference 2021, virtual event / ljubljana* (pp. 3432–3441). ACM / IW3C2, http://dx.doi.org/10.1145/3442381.3450111.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543).

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, Volume 1 (Long Papers)* (pp. 2227–2237).

Popat, K., Mukherjee, S., Yates, A., & Weikum, G. (2018). DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 22–32). Brussels, Belgium: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D18-1003, URL https://aclanthology.org/D18-1003.

Przybyla, P. (2020). Capturing the style of fake news. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34* (01), (pp. 490–497).

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 1–26.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*, 140:1–140:67.

Samadi, M., Mousavian, M., & Momtazi, S. (2021). Deep contextualized text representation and learning for fake news detection. *Information Processing & Management*, *58*(6), Article 102723. http://dx.doi.org/10.1016/j.ipm.2021.102723.

Schick, T., & Schütze, H. (2021a). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume* (pp. 255–269). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.eacl-main.20.

Schick, T., & Schütze, H. (2021b). It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies* (pp. 2339–2352). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.naacl-main.185.

Shen, W., Wang, J., & Han, J. (2014). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, *27*(2), 443–460.

Sheng, Q., Cao, J., Zhang, X., Li, R., Wang, D., & Zhu, Y. (2022). Zoom out and observe: News environment perception for fake news detection. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long Papers)* (pp. 4543–4556). Dublin, Ireland: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.acl-long.311.

Sheng, Q., Zhang, X., Cao, J., & Zhong, L. (2021). Integrating pattern-and fact-based fake news detection via model preference learning. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 1640–1650).

Shu, K., Cui, L., Wang, S., Lee, D., & Liu, H. (2019). Defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 395–405).

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, *8*(3), 171–188.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22–36.

Shu, K., Zheng, G., Li, Y., Mukherjee, S., Awadallah, A. H., Ruston, S., & Liu, H. (2020). Early detection of fake news with multi-source weak social supervision. In *Joint european conference on machine learning and knowledge discovery in databases* (pp. 650–666). Springer.

Silva, A., Han, Y., Luo, L., Karunasekera, S., & Leckie, C. (2021). Propagation2Vec: Embedding partial propagation networks for explainable fake news early detection. *Information Processing & Management*, *58*(5), Article 102618. http://dx.doi.org/10.1016/j.ipm.2021.102618.

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification? In M. Sun, X. Huang, H. Ji, Z. Liu, & Y. Liu (Eds.), *Lecture Notes in Computer Science*: *vol. 11856*, *Chinese computational linguistics - 18th china national conference* (pp. 194–206). Springer, http://dx.doi.org/10.1007/978-3-030-32381-3_16.

Sun, M., Zhang, X., Ma, J., & Liu, Y. (2021). Inconsistency matters: A knowledge-guided dual-inconsistency network for multi-modal rumor detection. In *Findings of the association for computational linguistics: EMNLP 2021* (pp. 1412–1423).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85.

Wang, Y., Ma, F., Wang, H., Jha, K., & Gao, J. (2021). Multimodal emergent fake news detection via meta neural process networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 3708–3716).

Wang, Y., Qian, S., Hu, J., Fang, Q., & Xu, C. (2020). Fake news detection via knowledge-driven multimodal graph convolutional networks. In *Proceedings of the 2020 international conference on multimedia retrieval* (pp. 540–547).

Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). DKN: Deep knowledge-aware network for news recommendation. In *WWW '18*, *Proceedings of the 2018 world wide web conference* (pp. 1835–1844). Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, http://dx.doi.org/10.1145/3178876.3186175.

Wei, X., Huang, H., Nie, L., Zhang, H., Mao, X.-L., & Chua, T.-S. (2017). I know what you want to express: Sentence element inference by incorporating external knowledge base. *IEEE Transactions on Knowledge and Data Engineering*, *29*(2), 344–358. http://dx.doi.org/10.1109/TKDE.2016.2622705.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., ... Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38–45). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.emnlp-demos.6.

Wu, L., Rao, Y., Yang, X., Wang, W., & Nazir, A. (2021). Evidence-aware hierarchical interactive attention networks for explainable claim verification. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 1388–1394).

Xu, W., Wu, J., Liu, Q., Wu, S., & Wang, L. (2022). Evidence-aware fake news detection with graph neural networks. In F. Laforest, R. Troncy, E. Simperl, D. Agarwal, A. Gionis, I. Herman, & L. Médini (Eds.), *WWW '22: The ACM web conference 2022* (pp. 2501–2510). ACM, http://dx.doi.org/10.1145/3485447.3512122.

Yang, H., Huang, S., Dai, X.-Y., & Chen, J. (2019). Fine-grained knowledge fusion for sequence labeling domain adaptation. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 4197–4206). Hong Kong, China: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D19-1429.

Yang, F., Liu, Y., Yu, X., & Yang, M. (2012). Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD workshop on mining data semantics* (pp. 1–7).

Yang, A., Wang, Q., Liu, J., Liu, K., Lyu, Y., Wu, H., She, Q., & Li, S. (2019). Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2346–2357). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P19-1226.

Zhang, X., Cao, J., Li, X., Sheng, Q., Zhong, L., & Shu, K. (2021). Mining dual emotion for fake news detection. In *Proceedings of the web conference 2021* (pp. 3465–3476).

Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, *57*(2), Article 102025. http://dx.doi.org/10.1016/j.ipm.2019.03.004.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: enhanced language representation with informative entities. In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics* (pp. 1441–1451). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/p19-1139.

Zhao, Y., Da, J., & Yan, J. (2021). Detecting health misinformation in online health communities: Incorporating behavioral features into machine learning based approaches. *Information Processing & Management*, *58*(1), Article 102390.

Zhao, G., Wu, J., Wang, D., & Li, T. (2016). Entity disambiguation to Wikipedia using collective ranking. *Information Processing & Management*, *52*(6), 1247–1257. http://dx.doi.org/10.1016/j.ipm.2016.06.002.

Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys*, *53*(5), 1–40.

Zhu, Y., Sheng, Q., Cao, J., Li, S., Wang, D., & Zhuang, F. (2022). Generalizing to the future: Mitigating entity bias in fake news detection. In E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, & G. Kazai (Eds.), *SIGIR '22: the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 2120–2125). ACM, http://dx.doi.org/10.1145/3477495.3531816.

Zubiaga, A., Liakata, M., & Procter, R. (2017). Exploiting context for rumour detection in social media. In *International conference on social informatics* (pp. 109–123). Springer.