Expert Systems With Applications 213 (2023) 118927

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Virtual prompt pre-training for prototype-based few-shot relation extraction

Kai He [a,b], Yucheng Huang [a,b], Rui Mao [c], Tieliang Gong [a,b], Chen Li [a,b], Erik Cambria [c,*]

[a] School of Computer Science and Technology, Xi'an Jiaotong University, China
[b] Shanxi Province Key Laboratory of Satellite and Terrestrial Network Technology Research and Development, China
[c] School of Computer Science and Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:
Few-shot learning
Information extraction
Prompt tuning
Pre-trained Language Model

ABSTRACT

Prompt tuning with pre-trained language models (PLM) has exhibited outstanding performance by reducing the gap between pre-training tasks and various downstream applications, which requires additional labor efforts in label word mappings and prompt template engineering. However, in a label intensive research domain, e.g., few-shot relation extraction (RE), manually defining label word mappings is particularly challenging, because the number of utilized relation label classes with complex relation names can be extremely large. Besides, the manual prompt development in natural language is subjective to individuals. To tackle these issues, we propose a virtual prompt pre-training method, projecting the virtual prompt to latent space, then fusing with PLM parameters. The pre-training is entity-relation-aware for RE, including the tasks of mask entity prediction, entity typing, distant supervised RE, and contrastive prompt pre-training. The proposed pre-training method can provide robust initialization for prompt encoding, while maintaining the interaction with the PLM. Furthermore, the virtual prompt can effectively avoid the labor efforts and the subjectivity issue in label word mapping and prompt template engineering. Our proposed prompt-based prototype network delivers a novel learning paradigm to model entities and relations via the probability distribution and Euclidean distance of the predictions of query instances and prototypes. The results indicate that our model yields an averaged accuracy gain of 4.21% on two few-shot datasets over strong RE baselines. Based on our proposed framework, our pre-trained model outperforms the strongest RE-related PLM by 6.52%.

1. Introduction

Relation Extraction (RE) is a fundamental task of data mining techniques, aiming to populate knowledge with facts from unstructured text. RE task means to extract the relations between two given entities. Many downstream applications rely on extracted relations, such as Information Retrieval (Guo et al., 2020), Question Answering (QA) (Lan & Jiang, 2021), and Knowledge Graph Construction (He, Yao, Zhang, Li, Li, et al., 2021). However, most existing supervised RE models (Wang, Fan, & Rose, 2020; Wang & Lu, 2020) are training with labeled data and face significant challenges in cross-domain processing. In contrast, few-shot learning only requires a small set of handful labeled examples, which has raised increasing attention in the research community alongside with zero-shot learning (Roy et al., 2022). For few-shot learning tasks, GPT-3 (Brown et al., 2020) proves the prominent ability for diverse task predictions by fusing a context and manual prompts without any further fine-tuning.

Inspired by this, some following studies (Lester, Al-Rfou, & Constant, 2021; Liu et al., 2021; Vu, Lester, Constant, Al-Rfou, & Cer, 2021) explore different methods to tune neural network models with prompts and obtain promising results. The main idea of prompt tuning is to reformulate various downstream applications as mask language prediction tasks. For example, given "I like this book. It is [MASK]"., the prompt tuning model learns the probabilities of "great" and "terrible" appearing in the [MASK] position to distinguish positive and negative sentiment polarities of the text ("I like this book".) before the prompt ("It is [MASK]".). Such an approach reduces the gap between the Pre-training Language Model (PLM) and downstream applications (Cambria, Liu, Decherchi, Xing, & Kwok, 2022; He, Mao, Gong, Li, & Cambria, 2022; Lin et al., 2021; Mao, Li, Ge and Cambria, 2022; Mao, Liu, He, Li and Cambria, 2022). Benefiting from the above advantages, prompt tuning becomes a popular technique in the low-data-resource research domains.

* Correspondence to: School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Ave, Block N4 #02a, Singapore 639798, Singapore.
E-mail addresses: hk52025804@stu.xjtu.edu.cn (K. He), huangyucheng@stu.xjtu.edu.cn (Y. Huang), rui.mao@ntu.edu.sg (R. Mao), gongtl@xjtu.edu.cn (T. Gong), cli@xjtu.edu.cn (C. Li), cambria@ntu.edu.sg (E. Cambria).

https://doi.org/10.1016/j.eswa.2022.118927
Received 18 July 2022; Received in revised form 20 September 2022; Accepted 25 September 2022
Available online 30 September 2022

Despite the great empirical success, prompt tuning still has two major limits: **(1)** Many prompt tuning works (Schick, Schmid, & Schütze, 2020; Schick & Schütze, 2021b) make an effort to manually create prompts. Handcrafting meaningful prompts is a brain-draining work, especially in the context of designing coherent prompts for extracting an abstract relation between two different entities; More importantly, the nuances in semantically similar natural language prompts may result in significant differences in model performance (Liu et al., 2021). To deal with the above problems, several automatic prompt generation models are proposed (Gao, Fisch, & Chen, 2021; Jiang, Xu, Araki, & Neubig, 2020; Shin, Razeghi, Logan IV, Wallace, & Singh, 2020). However, these methods generate discrete natural language prompts which are sub-optimal, because the generation process inevitably loses information that is learned in latent space. Other studies (Han, Zhao, Ding, Liu, & Sun, 2021; Shin et al., 2020) generated continual prompts with separated models.

These works cannot bridge the gap between PLM and downstream tasks, as the utilized PLM does not include the prompt-based training objects. Also, these work employs randomly initialized vectors as continual prompts. A robust initialization of prompt parameters of an employed PLM has not been paid enough attention in the community. **(2)** The prompt tuning model needs an additional process to map predicted words to label classes, which is named label word mappings. Noticeably, the selection of label words purely depends on empirical attempts. Most existing prompt tuning studies focus on text classification tasks, where label classes are not too many normally, such as positive or negative in sentiment analysis (Bao et al., 2021; Mao & Li, 2021). When the classification task comes to RE, the label space is much larger. In N-way-K-shot-based RE with FewRel 1.0 (Han et al., 2018) (our employed dataset), the number of relation classes reaches one hundred. Elaborated label word mappings for this task are manpower-costly and time-consuming. Besides, when the RE comes to a biomedical domain, it is hard to abstract an appropriate label word mapping to represent a specific relation name with a long sequence, such as "is normal tissue origin of disease" and "biological process involves gene product". Thus, such long sequence relation names are particularly challenging for prompt-tuning.

Motivated by the above limits, we propose a virtual prompt pre-training model (VPP) for prototype-based few-shot RE to omit the label words mapping and initialize more robust parameters for automatized prompts. There are two novel technical components in VPP: **(1)** We use a prompt-based prototype network to learn the relations between entities with a virtual prompt template. The prompt-based prototype network regards the probability distributions of label words in prompts as features, rather than typical hidden states from used neural models. It allows our model to take advantage of vocabulary-sized evidences (50,265 dimensions used in this paper) for predictions without introducing extra costs. It also reduces the gap between PLM and downstream tasks, and omits the label word mapping by comparing which prototype in the support set is the most similar to the instances in the related query set (N-way-K-shot setting is employed in this work). Since we use virtual prompts whose contexts are special tokens rather than natural language, the engineering of conventional prompts can also be omitted. **(2)** We propose an entity-relation-aware pre-training and a joint pre-trained prompt encoder for enhancing special tokens in our virtual prompts. There are four pre-training tasks, including mask entity prediction, entity typing, distant supervised RE, and contrastive prompt pre-training.

These pre-training tasks allow VPP to initialize effective parameters for PLM and a prompt encoder in few-shot learning RE. Compared with manual prompts or random initialized continual prompts by directly inserting special markers (Chen et al., 2021; Liu et al., 2021), our virtual prompt is initialized by a pre-trained encoder. Our joint pre-training further alleviates the gap between cold-start PLMs and prompt tuning, which can significantly boost performance.

We examine our method on few-shot learning tasks with two publicly available datasets (Gao et al., 2019; Han et al., 2018), demonstrating that VPP yields at least an average accuracy gain of 4.21% over strong external baselines (Gao et al., 2019; Peng et al., 2020; Qu, Gao, Xhonneux, & Tang, 2020; Snell, Swersky, & Zemel, 2017; Wang et al., 2021). Compared with the SOTA RE-related PLM proposed by the work (Peng et al., 2020), our model achieves 6.52% average gains, based on the same framework and virtual prompt tuning. Finally, our virtual prompt learning method exceeds manual prompts by at least 3.16%. The contribution of this work is summarized as threefold:

- We propose a virtual prompt-based prototype network that allows our model to omit the prompt template engineering in natural language and cumbersome label word mappings. It uses very high-dimension features to gain better discrimination in different label classes without introducing extra training costs.
- We propose an entity-relation-aware pre-trained model on the effectiveness of a PLM and a joint pre-trained prompt encoder, providing robust initialization for our virtual prompts. This approach further alleviates the gap between cold-start PLMs and prompt tuning.
- Our proposed model achieves outstanding performance in few-shot RE tasks. We conduct comprehensive experiments to analysis the improvements of proposed prompt-based joint pre-training and prototype network.

2. Related work

Few-Shot Relation Extraction. Generally, few-shot RE can be categorized into two classes. The former one seeks better representations through pre-training. KEPLER (Wang et al., 2021) integrated knowledge embeddings into PLMs by encoding textual entity descriptions and then jointly optimized the knowledge embeddings and language modeling objectives. The study (Peng et al., 2020) designed a contrastive relation pre-training object. The results demonstrated that task-specific pre-training could vastly improve the performance of related few-shot tasks. Another group explores different learning methods, based on existing PLMs. The study (Qu et al., 2020) proposed a Bayesian meta-learning method to learn the posterior distribution of the prototype vectors of relations, and parameterized it with a global relation graph for RE. MIML (Dong et al., 2020) employed a meta-information guided meta-learning method, taking advantage of semantic concepts of classes to enable more effective initialization and faster adaptation. Unlike these methods, our VPP simultaneously injects entity and relation knowledge by our proposed pre-training tasks, utilizes a different framework for few-shot predictions, and integrates the pre-training for continual prompts with PLMs.

Prompt Tuning. Two early studies manually constructed prompts for text classification (Schick et al., 2020; Schick & Schütze, 2021b). Manually constructed appropriate prompts are cumbersome and subjective. For such a reason, automated prompt creation methods were proposed. PTR (Han et al., 2021) applied logic rules to construct prompts, and tried to encode prior knowledge of each class into prompt tuning. However, these logic rules have to use specific entity types and rely on manual works. AutoPrompt (Shin et al., 2020) combined a set of trigger tokens according to a template with the original task input to create prompts, and employed a gradient-based search strategy to update them. BERTese (Haviv, Berant, & Globerson, 2021) adopted a paraphrasing-based approach to generate prompts. It converted an existing seed prompt to a collection of candidate prompts, and selected the ones with the best performance to use. The studies (Lester et al., 2021; Li & Liang, 2021) proposed lightweight alternatives for fine-tuning. They froze the parameters of PLM, and only updated a small task-specific vector as prompts. These parameter-freezing methods became competitive with typical fine-tuning methods, based on very large PLMs, e.g., T5 XXL that has more than 11 billion parameters (Raffel et al., 2020).

KnowPrompt (Chen et al., 2021) is similar to our work, which adapted prompt tuning in RE tasks. The difference is that KnowPrompt focused on injecting entity and relation information into generated prompts. Their prompts followed a fixed pattern, namely two entity type representations, concatenated with an extra [MASK]. However, their method heavily relies on external entity type information, and still needs manual mapping of label words to corresponding relations. Similar with PTR (Han et al., 2021), these methods cannot be employed in tasks that do not have entity type labels. Compared with the above studies using continual prompts (Chen et al., 2021; Han et al., 2021; Lester et al., 2021; Li & Liang, 2021; Shin et al., 2020), VPP focuses on joint pre-training a prompt encoder with PLM for more robust initialization in the scenario of few-shot RE tasks. Our method does not need any extra entity type information and any manual works.

3. Methodology

This section firstly introduces two technical components, namely the prompt-based prototype network and our joint pre-training method in general. Then, the corresponding details are described in Sections 3.1 and 3.2. A typical prompt tuning model predicts the most likely label word that appears in the [MASK] position, yielding relation labels with manually developed label word mapping rules. Alternatively, the prompt-based prototype network of VPP directly employs the probability distributions of [MASK] as features for calculating the spatial distances between query input and prototypes. The prototypes are calculated from the corresponding support set, where a prototype corresponds to a relation label that needs to be predicted. The final prediction is the relation label whose prototype is the most similar to the query instance. By such a comparison, VPP can avoid the arduous label word mappings. Additionally, the proposed prompt-based prototype network utilizes vocabulary-sized features (50,265 dimensions) to differentiate relation labels without any extra training costs. We argue these very high-dimension features inherently fit prompt-tuning, which allocate each word a probability to represent a weighted semantic meaning and offer great discrimination for classifying. Such features can be easily used in any prompt-tuning-based tasks, and our following experiments demonstrate their effectiveness.

Our virtual prompts used in the prompt-based prototype network are given by a template with special tokens, instead of a conventional prompt template in natural language. We believe that such a modification can avoid the semantic biases in prompt-tuning. Further, we pre-train the language model together with an additional prompt encoder to enhance these virtual prompts. The joint pre-training contains the tasks of Mask Entity prediction (ME), Entity Typing (ET), Distant supervised RE (DRE), and Contrastive Prompt pre-training (CP). The joint pre-training has two advantages: First, we can inject knowledge to obtain an entity-relation-aware PLM to better support our downstream tasks. Second, our pre-trained prompt encoder provides robust initialization for our virtual prompts and keeps the representations of input sentences and the prompts in the same semantic space. Basically, we use a pre-trained neural component to automatically generate context-aware prompts for different sentences, instead of previous works (Lester et al., 2021; Li & Liang, 2021; Liu et al., 2021) which use fixed and randomly initialized vectors.

3.1. A prompt-based prototype network

VPP defines each N-way-K-shot sample as a meta-task $\mathcal{M}$, which consists of a support set and a query set. In the support set, there are N relations, where each relation associates K sentences (see Section 4.1 for more details about N-way-K-shot, support and query sets). Thus, there are $N * K$ sentences with gold labels in the support set. In the query set, all sentences are with the same relation. The prompt-based prototype network should identify the relations between given entities of the query set within N relations from the support set.

The input instance $inst$ of the prototype network is given by

$$inst = Sent \oplus prompt, \tag{1}$$

where $\oplus$ indicates concatenation. An original sentence $Sent$ with special token insertions before and after each entity follows

$$\begin{aligned} Sent =& w_1, \dots, w_{a-2}, [Entity]_{a-1}, e^1_{a:b}, [/Entity]_{b+1}, w_{b+2}, \dots, \\ & w_{c-2}, [Entity]_{c-1}, e^2_{c:d}, [/Entity]_{d+1}, w_{d+2}, \dots, w_t, \end{aligned} \tag{2}$$

where [Entity] and [/Entity] denotes the start and end of entity tokens (e^1 and e^2), respectively. The subscript denotes the position of a token. $prompt$ in Eq. (1) is a prompt template, containing several special markers $[Pr]$, two entities (e^1 and e^2) and a $relation$ that e^1 and e^2 may be related.

$$\begin{aligned} prompt =& [Pr]_{\times n1} \oplus e^1 \oplus [Pr]_{\times n2} \oplus relation \\ & \oplus [Pr]_{\times n3} \oplus e^2 \oplus [Pr]_{\times n4} \oplus ? \oplus [MASK], \end{aligned} \tag{3}$$

where the hyper-parameters n_1, n_2 and n_3 are the numbers of the inserted special markers $[Pr]$ in different positions. $relation$ is the description of the relation label that needs to be predicted, such as "owned by" and "powered by". We introduce the symbol "?" in the prompt as a hint, aiming to indicate that the sequence before [MASK] of a prompt is a question. The prediction of [MASK] will be regarded as an answer. Thus, the whole process of prompt tuning simulates a masking language modeling task to reduce the gap between PLM and downstream tasks.

The traditional prompt is manually generated with natural language in discrete symbolic space (Schick et al., 2020; Schick & Schütze, 2021a). Some studies utilize the same vector with random initialization as a prompt for predicting all labels (Liu et al., 2021). In contrast, our utilized prompt is generated by the pre-trained prompt encoder. The benefit of using the prompt encoder is that generated prompts are continual and context-aware. The model can learn their features without biases in the choices of words of manually developed prompts. This idea is inspired by ELMo (Peters et al., 2018a) that uses a pre-trained neural component to obtain contextualized embeddings, rather than using fixed context-independent pre-trained vectors. The details of pre-training the prompt encoder are described in Section 3.2.

The process of predicting relations with the proposed prompt-based prototype network is illustrated as following. Given a sample $\mathcal{M}$, the input for each training step consists of N instances ($inst^q$) from a query set and $N * K * N$ instances ($inst^s$) from a support set. For the query input, an original query sentence ($Sent^q$) concatenates with a prompt with one of N different $relations$, forming $inst^q_j$, where $j \in \{1, \dots, N\}$ (see the pink box in Fig. 1). For the support input, $N * K$ original support sentences ($Sent^s_k$, where $k \in \{1, \dots, N * K\}$) respectively concatenate N $relation$, yielding $N * K * N$ instances ($inst^s_{k,l}$) (see the light blue boxes in Fig. 1).

In each training step, we feed the cluster of $inst^q$ and $inst^s$ into our proposed prompt-based prototype network to obtain the vocabulary-sized features for comparing. The vocabulary-sized features are represented by the probability distributions of [MASK], where the intuition is that we compare the responses of different relations with different prompts to get the final predictions. This method takes advantage of inherently generated probability distributions from the pre-training task of masking language modeling and provides larger discrimination without any extra training costs. In particular, we use the second-phase pre-trained PLM $Bart(\cdot)$ that consists of original Bart and an extra prompt encoder to obtain the presentations of a query cluster $inst^q$. Given a query input instance $inst^q_j$, the original Bart embedding layer and prompt encoder encode the sentence part and prompt part, respectively (see Fig. 2). The encoded presentations are concatenated and feed to the Bart, yielding the hidden state matrix for $inst^q_j$. The subscript of $Bart(\cdot)_{MASK}$ is introduced to denote the hidden state at the [MASK] position.

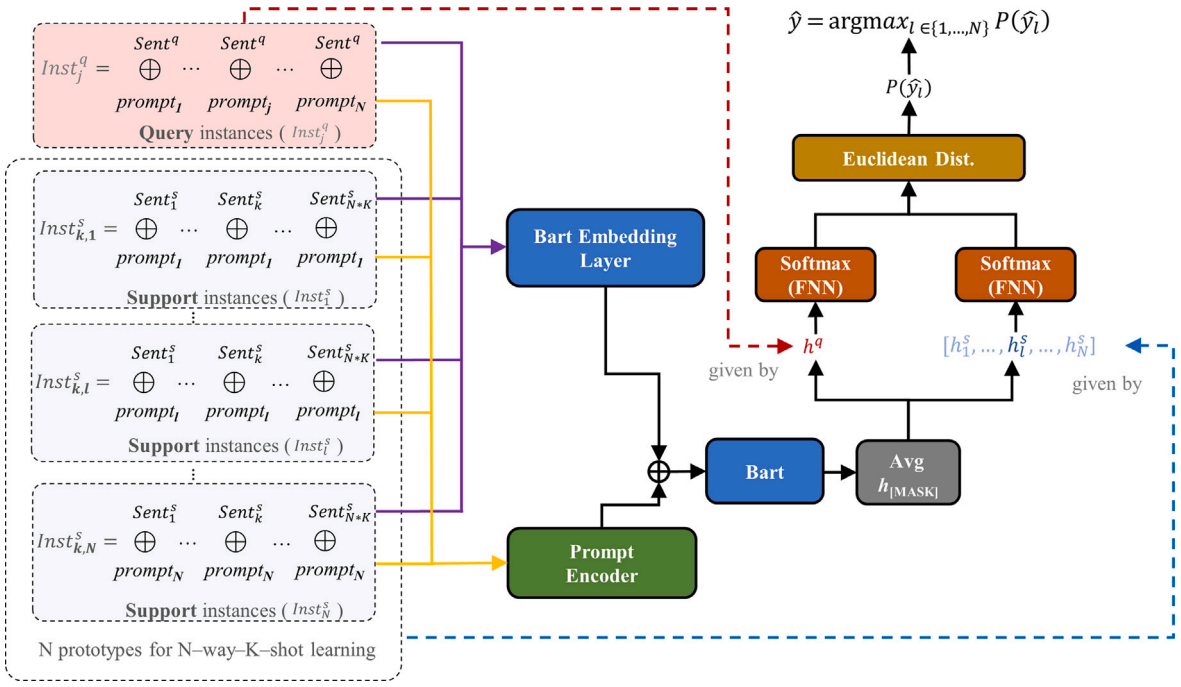

Fig. 1. The architecture of VPP. The gray box (Avg. $h_{[MASK]}$) denotes the averaged hidden states of the [MASK] tokens in prompts over $Inst_i^q, Inst_1^s, \dots, Inst_j^s, \dots, Inst_N^s$. $\oplus$ denotes concatenation. Our prompt encoder (the green box) consists of two Transformer layers. s and q denotes a support set and query set.

We average the [MASK] hidden states of the query sentence over N input instances, yielding the query representation h^q as:

$$h^q = \frac{1}{N}\sum_{j=1}^{N} Bart(inst_j^q)_{[MASK]}. \tag{4}$$

Then, a query masked word probability distribution vector $prob^q$ over the vocabulary of Bart is computed by

$$prob^q = Softmax(\omega * h^q + \epsilon), \tag{5}$$

where w and ϵ are pre-trained parameters in Bart. $prob^q$ is a vocabulary-sized feature of a relation for given entities in the query $Sent$. Different from using typical hidden states as features, each dimension of our features has a clear semantic meaning, which corresponds to a word in the vocabulary of Bart with chosen probability weighted. We do not have to use additional trainable parameters to learn the feature representations, thus, saving the costs of training.

Next, we feed the cluster of support instances $inst^s$ to the Bart model. The support representation of a prototype with $relation_l$ is

$$h_l^s = \frac{1}{N*K}\sum_{k=1}^{N*K} Bart(inst_{k,l}^s)_{[MASK]}. \tag{6}$$

A prototype masked word probability distribution vector $proto_l^s$ is

$$proto_l^s = Softmax(\omega * h_l^s + \epsilon). \tag{7}$$

$proto_l^s$ represents the answer feature of a $relation_l$ in $\mathcal{M}$. Each $\mathcal{M}$ contains N $proto^s$, corresponding N relations for predictions.

We compute the probability $P(\hat{y}_l)$ of the entities with $relation_l$ in the sentence $Sent^q$ of $inst^q$ by normalized Euclidean distance that is denoted as $d(\cdot)$.

$$P(\hat{y}_l) = \frac{exp(-d(prob^q, proto_l^s))}{\sum_{l=1}^{N} exp(-d(prob_i^q, proto_l^s))}. \tag{8}$$

Finally, the parameters of VPP are trained with cross-entropy loss. The predicted relation $\hat{y}$ for entity query $inst$ is

$$\hat{y} = arg \max_{l \in \{1,\dots,N\}} P(\hat{y}_l). \tag{9}$$

The method for computing prototype and query representations is sightly different. In Eq. (4), we average the [MASK] vectors of $inst^q$ with the same $Sent$ and different $relation_j$ ($j \in \{1, \dots, N\}$), yielding h^q. Alternately, in Eq. (6), we average the [MASK] vectors of $inst_l^s$ with the same $relation_l$ over different $Sent_k (k \in \{1, \dots, N * K\})$, yielding h_l^s as the representation of each prototype. We argue that Eq. (6)-based prototypes are more relation-centered to represent each class for support instances. It performs better than calculating sentence-centered representations by Eq. (4) in our experiments.

To sum up, we combine prompt tuning with a prototype network for few-shot relation extraction. The proposed model uses the representations of the clusters of $inst^q$ and $inst_l^s$ for calculating probability distributions of [MASK] markers over PLM's vocabulary, and compares these probability distributions to obtain the final predictions. It is significantly different from utilizing typical hidden states of input instances for gaining predictions of the most likely labels. Our method can alleviate the instability of presentations of $inst_l^s$ and $inst^q$, caused by insufficient training data in few-shot learning.

3.2. Joint pre-training for prompt encoder and entity-relation-aware PLM

Learning the qualified representations of entities and relations is important for RE. We design four pre-training tasks to improve entity and relation understanding of PLM. We also integrate a prompt encoder into the pre-training process for a more robust prompt tuning. Fig. 2 shows the framework of the pre-training model, input and output of each task. We first collect open-domain data from a Wikipedia dump,[1] labeling the entity type with NER tools (spaCy) automatically. The utilized biomedical data with entity information are token from the work (Xu et al., 2020). Its data resource comes from PubMed (Canese & Weis, 2013). Next, we employ distant supervision (Ji, Liu, He, & Zhao, 2017; Ren et al., 2017) to generate relation annotations by aligning with the knowledge base wiki-5M (Wang et al., 2021) and UMLS (Wheeler et al., 2007). We exclude sentences without any relations. All wiki-5M data are used for FewRel 1.0 pre-training.

[1] https://dumps.wikimedia.org/.

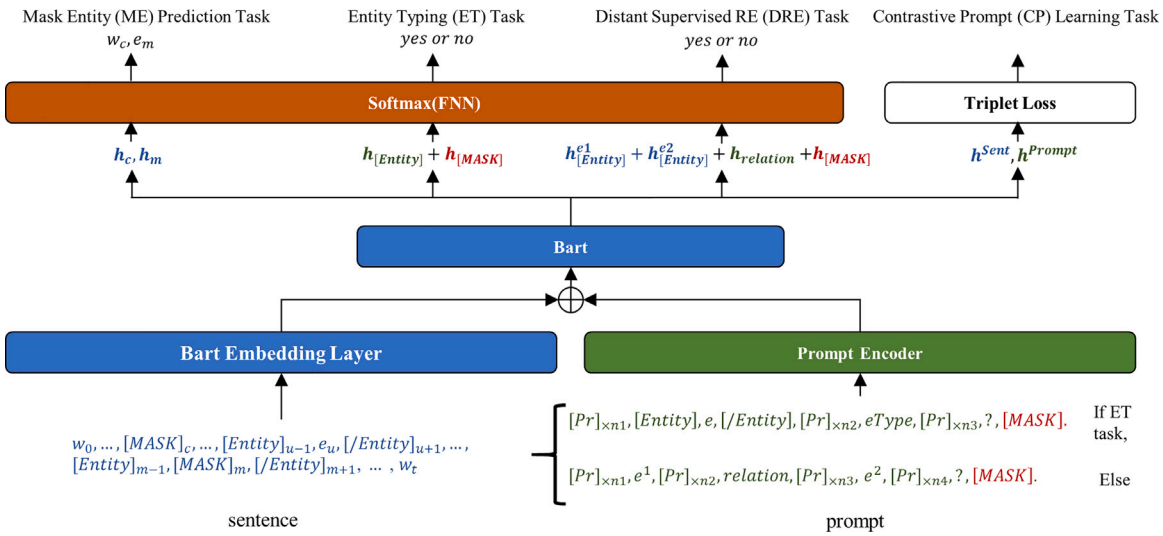

Fig. 2. The pre-training model and tasks. The subscript denotes a position. The colored boxes are in line with Fig. 1. The blue texts correspond to a sentence and its representations. The green texts correspond to a prompt and its representations. The red denotes [MASK] in the prompt and its corresponding representation. c denotes a masked word position in a context. u denotes the position of an unmasked entity (e). m denotes the position of a masked entity. $eType$ denotes the entity type of e. e^1 and e^2 are two random entities.

Half of wiki-5M combined with all UMLS are used for FewRel 2.0 pre-training. We have filtered out the overlap of the RE triples of the pre-training datasets from our evaluated FewRel 1.0 and FewRel 2.0 datasets. Thus, the pre-training data set is unbiased to the downstream RE evaluation tasks. Summarily, utilized pre-training data contains 19.5 million sentences, 166.63 million entities, and 230.77 million triples. More details about pre-training data can be found in our published data. The first pre-training task is a Masked Entity (ME) prediction task. The task means to predict masked tokens in the context of a sentence and masked entities. Given an input sentence, 10% single tokens in a context and 50% entities are randomly replaced with [MASK]. Thus, we use the following $Sent$ for pre-training to replace the $Sent$ in Eq. (2).

$$\begin{aligned} Sent = w_1, \dots, [MASK]_c, \dots, [Entity]_{u-1}, e_u, [/Entity]_{u+1}, \\ \dots, [Entity]_{m-1}, [MASK]_m, [/Entity]_{m+1}, \dots, w_t, \end{aligned} \tag{10}$$

where c denotes the position of a masked word in a context, u denotes the position of an unmasked entity (e_u), and m denotes the position of a masked entity. It is possible that more than two targets (tokens or entities) are masked out, if the sentence is very long. Given input $Sent$ and a prompt, our pre-training model tries to decode a single token or an entity at position r ($r \in R$, where $R = \{c, m, \dots\}$) by

$$p(\hat{y}^r_{ME}) \propto Exp(\omega_{ME} \cdot h_r + \epsilon_{ME}), \tag{11}$$

where $\hat{y}^r_{ME}$ is the sequence of decoded tokens, corresponding the [MASK] marker at position r of an input sentence. The ME pre-training loss is given by

$$\mathcal{L}_{ME} = -\sum_{r \in R} \log \prod_{w_o \in words} p(\hat{y}^r_{ME} | w_{\{o \le r-1\}}). \tag{12}$$

The second pre-training task is Entity Typing (ET) with prompts. The task means to identify if an entity type is correct or incorrect for an entity in a prompt. The inputs of ET are a $Sent$ and a prompt. At first, a randomly initialized prompt encoder takes a predefined prompt template $prompt^{ET}$ with an entity mention e and a sampled entity type $eType$ as input:

$$\begin{aligned} prompt^{ET} = &[Pr]_{\times n1} \oplus [Entity] \oplus e \oplus [/Entity] \oplus [Pr]_{\times n2} \\ &\oplus eType \oplus [Pr]_{\times n3} \oplus \text{“?”} \oplus [MASK]. \end{aligned} \tag{13}$$

Next, we randomly sample equal numbers of positive and negative $prompt^{ET}$, following the original sentence that contains the entity e as input. We denote a positive $prompt^{ET}$, if the sampled $eType$ is correct for the entity mention e in the prompt, otherwise negative.

Finally, after Bart decoding the sentence and prompt, the representations ($h_{[Entity]}$ and $h_{[MASK]}$) of [$Entity$], and [MASK] are summed up for predicting yes or no labels in the ET task.

$$p(\hat{y}^e_{ET}) \propto Exp(\omega_{ET} \cdot (h_{[Entity]} + h_{[MASK]}) + \epsilon_{ET}), \tag{14}$$

where the learnable parameters ω_{ET} and ϵ_{ET} are optimized with cross entropy loss $\mathcal{L}_{ET}$ in the ET task.

Distant supervised RE (DRE) is the third pre-training task. The task means to identify if a relation name in the prompt is correct or incorrect for two entities in a sentence. Given a set of labeled entities in a sentence, VPP combines any two entities as a pair for RE. We automatically annotate the relation between an entity pair in a distant supervised way by using wiki-5 m and UMLS knowledge base. We define positive and negative DRE prompts ($prompt+, prompt-$) by inserting correct and incorrect relations in the prompt template in Eq. (3). The output is yes or no labels. The prediction $\hat{y}^{e1,e2}_{DRE}$ of an entity pair (e^1, e^2) is given by

$$\begin{aligned} p(\hat{y}^{e1,e2}_{DRE}) \propto &Exp(\omega_{DRE} \cdot (h^{e1}_{[Entity]} + h^{e2}_{[Entity]} \\ &+ h_{relation} + h_{[MASK]}) + \epsilon_{DRE}), \end{aligned} \tag{15}$$

where $h^{e1}_{[Entity]}$, $h^{e2}_{[Entity]}$, and $h_{relation}$ are the representations of [$Entity$] for e^1, [$Entity$] for e^2, and $relation$. [$Entity$] tokens are in the sentence part, while $relation$ is in the prompt part. We employ cross entropy loss $\mathcal{L}_{DRE}$ for learning DRE.

The last pre-training task is Contrastive Prompt pre-training (CP). We employ a contrastive triplet loss (Balntas, Riba, Ponsa, & Mikolajczyk, 2016), aiming at learning representations by pulling instances with similar meanings together and pushing instances with different meanings apart in latent space. CP task has three input sequences: (1) a masked sentence (Eq. (10)), (2) the masked sentence and a positive prompt, (3) the masked sentence and a negative prompt. The prompt template (Eq. (3) or Eq. (13)) is defined by a switching function (see Eq. (18) later) between ET and DRE. We first encode the masked sentence alone to obtain the representation h^{Sent}. Next, the masked sentence and a positive prompt are fed to VPP, yielding $h^{prompt+}$. Then, we feed the masked sentence and a negative prompt to obtain $h^{prompt-}$. These representations are used for computing the CP loss $\mathcal{L}_{CP}$

$$\begin{aligned} \mathcal{L}_{CP} = &max\{\|h^{Sent} - h^{prompt+}\|_2 - \\ &\|h^{Sentence} - h^{prompt-}\|_2 + \mu, 0\}, \end{aligned} \tag{16}$$

Table 1
An example for a 2-way 1-shot scenario. Bold text denotes entities. For DA challenge of FewRel 2.0, samples in the training and testing sets come from different domains.

	Relation class	Training set (Wiki data)
Support set	(A) locate in (B) founded by	The airline's hub is **Maya Airport** in **Brazzaville**. **Steve Jobs** was the chairman, and co-founder of **Apple Inc**.
Query set	(A) or (B)	Nearest airport is **Kazi Nazrul Islam Airport**, **Durgapur**.
	Relation class	Test set (Biomedicine data)
Support set	(A) ingredient of (B) gene in organism	... effects of **oxybutynin chloride** with cellulose (modified **oxybutynin**). ... an anatomical map of the **human** a **syn** distribution in aso mice.
Query set	(A) or (B)	... mir-k12-11, an ortholog of the **human tumor** gene **hsa-mir-15**.

where μ is a hyper-parameter as a set margin. Finally, the total loss $\mathcal{L}_{total}$ is formulated as:

$$\mathcal{L}_{total} = \lambda_{ME}\mathcal{L}_{ME} + \lambda_{CP}\mathcal{L}_{CP} + \lambda_{ET}\mathcal{L}_{ET} \cdot I(\gamma) + \lambda_{DRE}\mathcal{L}_{DRE} \cdot (1 - I(\gamma)) \tag{17}$$

where $\lambda_{ME}, \lambda_{ET}, \lambda_{CP}, \lambda_{DRE}$ are the weights of losses for allocating the significance of each tasks. γ is a random number between 0 and 1. $I(\cdot)$ is a switching function for learning RE and ET, given by:

$$I(\gamma) = \begin{cases} 1, & if\ \gamma \le \beta \\ 0, & if\ \gamma > \beta \end{cases} \tag{18}$$

where β is a hyper-parameter for controlling the ratio of pre-training ET and DRE tasks.

The pipeline of the pre-training is: (1) we select ET or DRE pre-training task according to Eq. (18); (2) positive and negative *prompts* are concatenated with *Sent*; (3) ME receives *Sent* as inputs; DRE or ET regards (*Sent*, *prompts*+) and (*Sent*, *prompts*−) as inputs; CP learns from the triplet (*Sent*, *prompts*+, *prompts*−).

The intuition of introducing the proposed joint pre-training tasks can be summarized as follows: First, it can be regarded as a second phase pre-trained for injecting knowledge to the original Bart. Inspired by previous studies (Sun et al., 2021, 2020), we design ME, ET, and DRE tasks to achieve an entity-relation-aware PLM with more understanding of entities, entity types, and relations. These understandings are also helpful for downstream RE learnings. Second, the last pre-training task CP provides prompt tuning with a prompt encoder. This prompt encoder generates initialized and context-aware prompts for prompt tuning, instead of manual or random parameter-initialized prompts. This pre-training task ensures the employed prompts in the same semantic space and further alleviates the requirement of annotated data. The following experiments demonstrate that prompt tuning is significantly benefited from our joint pre-training for the prompt encoder and the entity-relation-aware PLM.

4. Experiments

4.1. Experiment settings

Formulation of N-way-K-shot. In this work, we focus on N-way-K-shot RE tasks. We first divide the whole dataset into training, validation, and testing sets. There are no overlapped relation types among them. The training, validation, and testing sets are divided into the pairs of support sets and query sets. A support set contains N relation name classes, randomly sampled from all corpus. Each class has K instances. A query set contains arbitrary instances for predictions. The included relation name classes of instances in a query set should be the subset of its corresponding support set. 5-way-1-shot, 5-way-5-shot, 10-way-1-shot, and 10-way-5-shot are four setups in our experiments.

Dataset. Following baseline works (Gao et al., 2019; Peng et al., 2020; Qu et al., 2020; Wang et al., 2021), we evaluate VPP on the FewRel 1.0 (Han et al., 2018) and FewRel 2.0 (Gao et al., 2019), which following the N-way-K-shot setups. FewRel 1.0 only focuses on few-shot RE. Its training, validation, and testing sets are sourced from wiki data. FewRel 2.0 proposed a few-shot Domain Adaptation (DA) challenge, which aims to evaluate across-domain abilities of few-shot learning models. Its validation and testing set data come from a medical domain, while the train set data are in a general domain. The example of 2-way-1-shot for the DA task is shown in Table 1. The detailed dataset statistics are shown in Table 2.

Baselines. (1) Prototype network (Snell et al., 2017) utilizes all related instances in the support set to calculate a prototype for each relation class. It compares the distance between instances in a query set with these prototypes for predictions. **(2) Pair network** is based on the sequence classification model, which is proposed in FewRel 2.0 (Gao et al., 2019). It pairs and concatenates each query sentence with all supporting sentences, then employs a sequence classification model to predict if the two instances express the same relation. **(3) REGRAB** (Qu et al., 2020) is a novel Bayesian meta-learning method to effectively learn the posterior distribution of the prototype vectors of relations, where the initial prior of the prototype vectors are parameterized with a graph neural network on the global relation graph. **(4) KEPLER** (Wang et al., 2021) is a unified frame for knowledge embedding and pre-trained language representations, which jointly optimizes the knowledge embedding and language modeling objectives. **(5) Proto-CP** (Peng et al., 2020) focuses on studying the effects of textual contexts and entity mentions in relation extraction, based on conventional prototype network. It uses an entity-masked contrastive pre-training framework for RE to gain a deeper understanding of the above two factors. **(6) TPN** (Wen, Liu, Ouyang, Lin, & Chung, 2021) integrates the transformer model into a prototypical network for more powerful relation-level feature extraction, and focuses on sequence learning without catastrophic forgetting. **(7) ConceptFERE** (Yang, Zhang, Niu, Zhao, & Pu, 2021) designs an attentive model to measure each word for a specific class and introduces the inherent concepts of entities to provide clues for relation prediction. **(8) Prefix-Tuning** (Li & Liang, 2021) is a prompt-based method, which keeps language model parameters frozen, while tunes small continuous embeddings for downstream tasks. **(9) KnowPrompt** (Chen et al., 2021) is another prompt-based method. This method jointly tunes continuous prompt and label words to inject entity type information into used prompts. These prompts and label words are randomly initialized. **(10) PRT** (Han et al., 2021) applies logic rules to construct prompts and also try to inject prior knowledge into constructed prompts. This method is also influenced by label word mapping. We cannot compare with AutoPrompt (Shin et al., 2020), this study needs lots of relation-specific annotated data to train prompts, so it also cannot be used in the few-shot RE tasks. All the above prompt-based methods need label words mapping, while proposed VPP takes advantage of the comparison of concatenated prompts to take place label word mapping.

Evaluation and Hyper-parameters. By using the official evaluation website[2] for FewRel 1.0 and FewRel 2.0, we report performances measured by averaged accuracy over 10,000 testing instances in each N-way-K-shot setup. VPP utilizes base-BART with 768 hidden dimensions. The maximum length of input sentences is 128. Adam

[2] https://thunlp.github.io/fewrel.html.

Table 2
Statistics of FewRel 1.0 and FewRel 2.0 benchmark. $test\mathcal{M}$ means the number of N-way-k-shot instance combinations for reporting performances.

Corpus		# Relation	# Entity	# Instance	# $test\mathcal{M}$
Common	Training	64	89,600	44,800	–
1.0	Validation	16	22,400	11,200	–
	Test	20	28,000	14,000	10,000
2.0	Validation	10	2000	1000	–
	Test	15	3000	1500	10,000

optimizer (Kingma & Ba, 2017) is employed with the initial learning rate of 2e-5. λ_{ME}, λ_{CP}, λ_{ET}, λ_{DRE} in Eq. (17) are 1, 1, and 2, 2, respectively. The more details of hyper-parameters can be found in released codes.

4.2. Main results

We consider the averaged accuracy over different N-way-K-shot setups as the main measure for benchmarking. We manually create two baseline prompts (-M_1 and -M_2) in natural language with different styles for each relation in FewRel 1.0 and FewRel 2.0 datasets to compare with the proposed continual prompts. Taking the relation "located next to" as an example, manual Prompt 1 (-M_1) is designed as a question style "Does Entity_1 located next to Entity_2? [MASK]", and manual Prompt 2 (-M_2) is following a declarative pattern as "Entity_1 adjoin Entity_2 [MASK]". FewRel 2.0 task is more difficult than FewRel 1.0, because it poses DA challenges. Proto and Pair are conventional prototype network (Snell et al., 2017) and pair network (Gao et al., 2019) baselines in the result tables. Compared with external baselines, our proposed VPP-JP-C (JP denotes our joint pre-training, C denotes our continual prompt) exceeds the strongest baseline (Proto-CP) by 4.21% accuracy on average. Noticeably, in the DA challenge, the advantage of our model over Proto-CP significantly raises accuracy to 8.12%. It shows that our method is more effective in cross-domain few-shot RE. Additionally, our model achieves the best performance on 7 out of 8 N-way-K-shot evaluation tasks. It also shows the utility of our model in few-shot learning.

Next, we compare our proposed Joint Pre-trained (JP) model with different PLMs (-Bert, -Bart, -KEPLER, -CP) in the same prompt-based prototype network (VPP-) and different prompts (-M_1 and -M_2). For the manual prompt-based models (VPP-M_1), VPP-JP-M_1 outperforms the second best PLM (VPP-CP-M_1) by 2.56%. For the continual prompt-based models (VPP-C), VPP-JP-C outperforms the second best PLM (VPP-CP-C) by 6.52%. Thus, our proposed JP model is more supportive in our prompt-based prototype network. We do not embed JP in Pair- and Proto-frameworks for benchmarking, because they are not prompt tuning-based methods. Compared with existing prompt-based method (Chen et al., 2021; Han et al., 2021; Li & Liang, 2021), our VPP-JP-C outperform than these methods significantly. Prefix-Tuning (Li & Liang, 2021) freezes used PLM and only fine-tunes a few new parameters introduced by using continuous prompts. The results show that this method cannot handle the DA task very well in FewRel 2.0 dataset. PTRd (Han et al., 2021) and Knowprompt (Chen et al., 2021) are two similar methods, which use randomly initialized vectors as prompts and fine-tune all their parameters. Compared with other methods which use specific PLMs enhanced by entity and relation information, such methods have no advantages.

We also evaluate the improvements of our prompt tuning method (VPP-) by controlling PLMs. In Bert based models, VPP-Bert with a manual prompt (VPP-Bert-M_1) significantly outperforms Proto-Bert by 7.55% and 31.65% in FewRel 1.0 and 2.0 datasets. With continual prompts, VPP-Bert-C surpasses Proto-Bert by 22.26% overall. Our prompt-based method presents improvements when we utilize another secondary PLM, e.g., KEPLER. VPP-KEPLE-M_1 yields a general gain of 3.64% over Proto-KEPLER.

Noticeably, the joint pre-training significantly boosts model performance compared with the random initialization of the prompt encoder. We randomly initialize the green box (prompt encoder) in Figs. 1 and 2 (VPP-JP-C_{RI}) and keeping the rest same as VPP-JP-C. We observe that the overall averaged accuracy of VPP-JP-C_{RI} is 3.13% lower than VPP-JP-C, which signifies the importance of the joint pre-training.

Finally, we compare our continual prompt (-C) with manual prompts (-M_1 and -M_2) in the same framework (VPP-JP-). The two manual prompt-based models yield different performance with a gap of 1.96% with -JP. It supports the finding of the research (Liu et al., 2021) that the nuance in manual prompts may result in sharp differences in accuracy. However, our proposed virtual continual prompt can mitigate the variations of prompts, because the context of the virtual continual prompt is based on the same special token ([Pr]), rather than natural language. The only hyper-parameters in the virtual prompt is the number of special tokens, which is tested later. Such a prompt-tuning paradigm (VPP-JP-C) brings extra gains of 3.16% in average accuracy over the best manual prompt (VPP-JP-M_1).

5. Discussion

Why does our continuous prompt outperform discrete prompts and existing continuous prompts?

First, we discuss why continuous prompt outperform discrete prompts. It should notice that the proposed prompt encoder contains a certain amount of pre-training parameters. These parameters enable VPP to output different prompts for different contexts accordingly, even with the same relations. In particular, both a triple (entity1, relation, entity2) and its different context decide the prompt representation after PLM encoding. It is significantly different from typical prompt tuning studies, which usually utilize an unchanged prompt for each class. Besides, all generated continual prompts of VPP take virtual markers [Pr] as parts of inputs. These virtual markers do not have any specific semantics. Then, the representations of the virtual markers that are trained with the associated contexts are unbiased representations. Noticeably, prompt generation in natural language may lose latent information. PLM usually carries out a LogSoftMax operation on the continual representations and yields the most likely lexical sequences (discrete prompts). These tokens will be then encoded again in latent space for downstream task learning, yielding new representations that are different from the representations in the previous step. To this end, the discrete prompts experience separated decoding and encoding processes towards the same lexical sequences, while our continual prompts can optimize the prompt representations globally from end to end to achieve accurate predictions.

Second, we discuss the difference between VPP and existing continuous prompts studies. The two core differences are our virtual prompts are pre-trained and we do not need label words mapping to achieve classifications. Specifically, studies (Lester et al., 2021; Li & Liang, 2021) freeze the parameters of employed PLM to tune continuous prompts, which means they aim to train a small number of parameters from scratch to adapt downstream tasks. The benefit of such methods is they are training vastly fewer parameters than fine-tuning based methods, to perform well in few-shot learning. The limit is that the frozen PLM parameters cannot adapt to downstream task domains, e.g., biomedical domains, during fine-tuning. Only a few parameters of their model (the parameters that do not belong to a PLM) are fine-tuned on downstream tasks. Thus, the overall framework is not well fine-tuned. The studies (Chen et al., 2021; Han et al., 2021)are similar to our VPP, which employ continuous vectors as prompts and fine-tune all parameters of a PLM. The difference is that our continuous prompts are from a prompt encoder, which is jointly pre-trained with the used PLM, while randomly initialized embedding is used in these compared baselines.

Table 3
Accuracy (%) on the testing sets of FewRel 1.0 and FewRel 2.0 Domain adaption (DA) challenge. Proto, Pair, and VPP mean using conventional prototype network (Snell et al., 2017), pair network (Gao et al., 2019) and our proposed model; -Bert, -Bart, -KEPLER, and -JP mean using Bert, Bart, KEPLER, and our PLM as the backbone, respectively; -M_1 and -M_2 mean using two sets of manual prompts; -C and -C_{RI} mean continual prompts generated by joint-pre-trained or randomly initialized prompt encoder. To fair comparison, we use Bart-base for these baselines.

Model	FewRel 1.0					FewRel 2.0 (DA)					Avg. (All)
	5-1	5-5	10-1	10-5	Avg. (1.0)	5-1	5-5	10-1	10-5	Avg. (2.0)	
Proto-Bert	80.68	89.60	71.48	82.89	81.16	40.12	51.50	26.45	36.93	38.75	59.96
Pair-Bert	88.32	93.22	80.63	87.02	87.30	67.41	78.57	54.89	66.85	66.93	77.12
TPN	80.14	93.60	72.67	89.83	84.06	60.35	81.60	38.12	76.91	64.25	74.16
ConceptFERE	89.21	93.98	75.72	86.21	86.28	–	–	–	–	–	–
REGRAB	90.30	94.25	84.09	88.20	89.21	–	–	–	–	–	–
Prefix-Tuning[a]	82.18	91.46	75.67	88.11	84.36	57.14	67.12	52.00	58.93	58.80	71.58
PTR[a]	89.42	92.03	84.00	88.51	88.49	63.55	83.12	54.05	71.45	68.04	78.70
Knowprompt[a]	90.12	94.23	85.97	89.62	89.99	61.22	82.01	55.45	72.01	67.67	78.83
Proto-KEPLER	88.30	95.94	81.10	92.67	89.50	66.41	84.02	51.85	73.60	68.97	79.24
Pair-KEPLER	90.31	94.28	85.48	90.51	90.14	67.23	82.09	54.32	71.01	68.66	79.40
Proto-CP	95.10	97.10	**91.20**	94.70	94.50	79.70	84.90	68.10	79.80	78.12	86.31
VPP-Bert-M_1	87.83	95.10	82.81	89.11	88.71	68.74	85.03	55.71	72.11	70.40	79.56
VPP-Bert-C	87.34	94.48	82.91	88.17	88.22	70.29	89.99	58.84	85.71	76.21	82.22
VPP-Bart-M_1	89.26	94.42	82.50	88.12	88.58	68.48	84.67	56.50	73.55	70.80	79.69
VPP-Bart-C	89.17	93.32	81.85	85.31	87.42	64.49	76.20	50.09	67.70	64.62	76.02
VPP-KEPLER-M_1	91.02	96.06	84.15	90.03	90.32	73.84	89.88	59.52	78.81	75.51	82.92
VPP-KEPLER-C	88.90	95.37	84.59	88.64	89.38	71.68	88.50	59.85	78.02	74.51	81.95
VPP-CP-M_1	92.74	96.89	88.80	90.01	92.11	75.97	89.59	63.32	81.05	77.48	84.80
VPP-CP-C	91.58	95.42	88.82	89.64	91.37	75.21	88.51	61.99	80.86	76.64	84.00
VPP-JP-M_1	92.82	96.70	88.39	92.45	92.59	81.49	91.11	70.03	85.89	82.13	87.36
VPP-JP-M_2	90.57	94.67	86.15	90.26	90.41	79.45	90.02	68.84	83.21	80.38	85.40
VPP-JP-C_{RI}	91.01	95.45	88.01	90.88	91.34	83.15	91.04	74.30	85.28	83.44	87.39
VPP-JP-C	**95.32**	**97.84**	90.08	**95.96**	**94.80**	**86.78**	**93.04**	**77.41**	**87.71**	**86.24**	**90.52**

[a]Means the results are from our repetition.

Table 4
The proposed VPP-JP-C model performance, given by different patterns (Pat_1, Pat_2 and Pat_3) of prompts and different hyper-parameters of n_1, n_2, n_3, n_4 in Eq. (3). The reported performance is measured by accuracy on the 5-way-1-shot validation set of FewRel 2.0.

5–1	$[n_1, n_2, n_3, n_4]$	Pat_1	Pat_2	Pat_3	Avg.
VPP-JP-C	[1, 1, 1, 1]	85.40	83.92	85.30	84.87
	[2, 2, 2, 2]	86.03	82.80	85.40	84.74
	[1, 3, 3, 1]	88.00	84.24	**88.05**	86.76
	[1, 5, 5, 1]	87.62	84.11	86.91	86.21
	[3, 3, 3, 3]	87.58	83.95	87.52	86.35
	[5, 5, 5, 5]	87.40	84.30	87.60	86.43
VPP-Bart-M	–	M_1	M_2	–	
	–	69.75	62.13	–	65.94

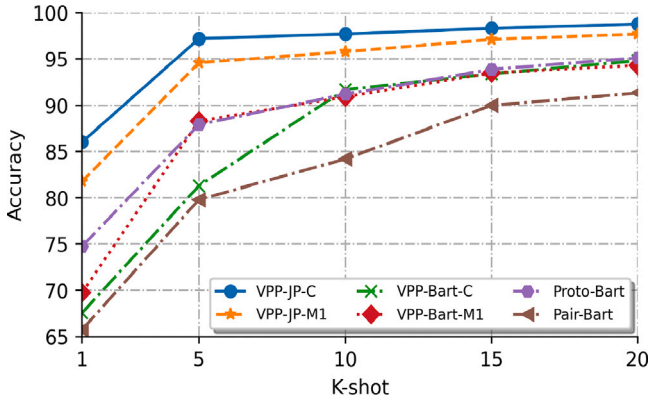

Fig. 3. The comparison of continual prompts, manual prompts, original BART and our joint pre-training, and baseline models with different numbers of training instances (K-shot). The reported accuracy is on the validation set of 5-way-K-shot of FewRel 2.0.

This improvement is inspired by ELMo (Peters et al., 2018b), which first proposes that we can employ a pre-trained neural structure to generate contextualized word representations, rather than using fixed word embedding like word2vector (Mikolov, Chen, Corrado, & Dean, 2013). With such a pre-trained prompt encoder, our continuous prompts can be contextualized. Also, we can update all parameters of used PLM to adopt a new domain, and meaningful continuous prompt initialization intuitively helps the model achieve better performance, specifically under the few-shot setting. By comparing VPP-Bart-C and VPP-JP-C in Table 3, our pre-trained prompt is much better than random initialized continuous prompts with 7.38% F1 improvements. We conclude that not only pre-training can benefit language models, but also benefit methods that need automatically generated prompts.

Does the pattern of prompts matter model performance? The work (Liu et al., 2021) demonstrated that using different manual prompts on the same instance results in a 19.79% P@1 measure gap. Thus, we explore the impact of a prompt with different patterns (namely different numbers and positions for the inserted [Pr] markers). We use different hyper-parameters for $[n_1, n_2, n_3, n_4]$ of Eq. (3), such as [1,1,1,1], [2,2,2,2], [1,3,3,1], [1,5,5,1], [3,3,3,3] and [5,5,5,5].

We also examine different prompt patterns (Pat_1 and Pat_2), apart from the recommended pattern (Pat_3) that is defined in Eq. (3).

$$Pat_1 = [Pr]_{\times n1} \oplus e^1 \oplus [Pr]_{\times n2} \oplus e^2 \oplus [Pr]_{\times n3} \oplus relation\_name \oplus [Pr]_{\times n4} \oplus ? \oplus [MASK],$$

$$Pat_2 = [Pr]_{\times n1} \oplus relation\_name \oplus [Pr]_{\times n2} \oplus e^1 \oplus [Pr]_{\times n3} \oplus e^2 \oplus [Pr]_{\times n4} \oplus ? \oplus [MASK].$$

Finally, our continual prompt performance (VPP-JP-C) with different setups in parameters and patterns benchmarks with manual prompts (VPP-Ba-M_1 and VPP-Ba-M_2). As shown in Table 4, the recommended setup of $[n_1, n_2, n_3, n_4]$ = [1,3,3,1] and Pat_3 yields the highest accuracy (88.05%) on the FewRel 2.0 validation set. Generally, a longer prompt with more [Pr] special token insertions likely yields better performance than the short ones.

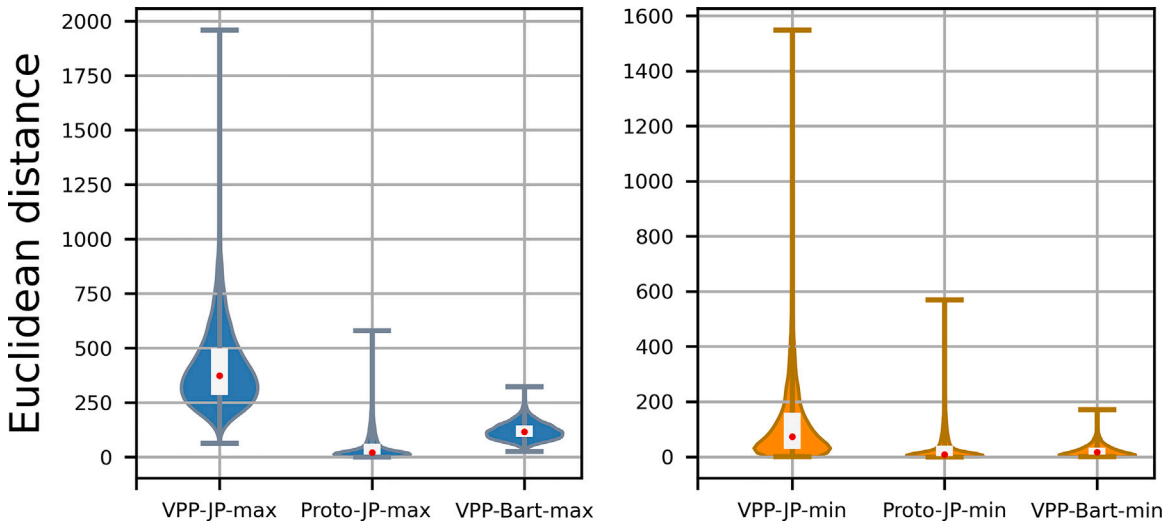

Fig. 4. The comparison between the classes of the query set and the most similar (-min) and different (-max) classes in the support set in Euclidean distance space, given by different models. The Euclidean distance is given by 10-way-1-shot of FewRel 2.0 validation set.

We also observe that although different patterns and different hyper-parameters in n result in different performance, the gap between different setups of the continual prompt-based model in the same column or the same row is much smaller than the gap (7.62%) between two manual prompt-based models (M_1 and M_2). VPP-JP-C also outperforms VPP-Bart-M on average with different setups. This shows that our proposed model yields robust performance with limited variations.

How does the number of training data effect prompt tuning? Considering the advantages of prompt tuning for few-shot learning tasks, we compared different methods against VPP by increasing the number of training data (K shot). As shown in Fig. 3, all methods benefit from learning more shots on FewRel 2.0. Apart from the Pair network with Bart (Pair-Bart), other few-shot learning models achieve apparent performance improvements when the shot number rises from 1 to 5, and the improvements become slower when the number of shots keeps increasing. The biggest advantage of VPP-JP-C against other baseline models appears in 5-way-1-shot and 5-way-5-shot. This clearly demonstrates the strength of our proposed method. Additionally, in the 5-way-20-shot evaluation task, our proposed model also achieves the best performance, although the improvement of our method is not exciting after 5-shot. It shows that the model still has strong performance on many-shot learning. In contrast, prototype and pair networks (Proto-Bart and Pair-Bart) have sharper slop than VPP-JP-C after 5-shot. The last but not the least, our entity-relation-aware joint pre-training (JP) also alleviates the annotation dependence for downstream RE tasks to some extent, when we compare the gaps between VPP-JP and VPP-Bart models.

Why does VPP outperform typical prototype network? The most significant difference between VPP and a typical prototype network is the inputs for calculating distances between query instances and the prototypes of different classes. In VPP, we employ the predicted probability distribution of the [MASK] token, instead of PLM hidden states that are employed by a conventional prototype network for RE classification. Since the vocabulary size of a PLM is very big (the vocabulary size of Bart-base is 50,265), VPP can achieve more distinguishable features for decision. As shown in Fig. 4, such a method can differentiate the classes of $\mathcal{M}$ better than a conventional prototype-based method with a large margin by comparing VPP- with Proto-. Without our proposed joint pre-training (JP), the euclidean distances of the VPP model, calculated from VPP-Bart- still yield better discrimination, compared with VPP-JP-.

Usability and Limitations. The proposed VPP contains two main components, i.e. the prompt-based prototype network and our joint pre-training PLM. Because these two components are jointly pre-trained, they can compatibly work together. Further, the prompt-based prototype network can be separately used as a prompt-based method. The advantage of this component is that it has no manual prompt construction and label words mapping. The limitation of this prototype network is that we need to match each sentence with N virtual prompts, which means we need N times batch sizes or N times costs. For the second component, our joint pre-training PLM can be separately employed with the first component. The prompt encoder can separately utilized for generating prompts as well. Our PLM can support many downstream tasks which need to understand entities and relations, such as intention recognition in question answering and dialog system. In addition, the prompt encoder, which was jointly pre-trained with BART, is regarded as a small pre-trained model. It can support other studies which need automatically generated prompts.

There are another two limitations in VPP. First, in order to achieve better domain adaptation in downstream tasks, VPP fine-tunes all its parameters, rather than typical prompt tuning methods that simply fine-tune a small portion of parameters, e.g., the parameters for learning prompts (Lester et al., 2021; Li & Liang, 2021). Thus, the computational cost of VPP is higher than that of those typical prompt tuning methods. The size of the proposed VPP is almost equal to BART-base, while the only extra parameters come from the prompted encoder (two Transformer layers). In our experiments, GPU memory consumption for fine-tuning all the parameters is about 2.21 times that of just fine-tuning parameters for the used prompts. However, time costs are not very different in fine-tuning all or partial parameters, because neural models have to back propagate through all weights to compute gradients. In our experiments, the time cost of VPP is about 1.13 times that of typical prompt tuning. Another limitation of VPP is that our prompt encoder was jointly pre-trained with BART. It is hard to ensure the learned semantic space is compatible with other PLMs. Thus, using a different PLM instead of BART may cause a drop in accuracy.

6. Conclusion

This paper proposes a virtual prompt pre-training model, which expands prompt tuning to few-shot RE tasks. The proposed model utilizes continual prompts that are automatically generated from a pre-trained prompt encoder, which provides robust initialization for used prompts to replace random vectors. By using a virtual prompt template, our model eliminates the labor-intensive label word mappings in tasks with a large label space. It is also a practical method for the prompt-tuning-based classification that cannot manually generate coherent label word mappings for long label sequences with rich semantics.

Finally, our proposed pre-training tasks deliver sufficient prior-knowledge for jointly initializing the prompt encoder and Bart. We demonstrate that our query-prototype modeling method which utilizes the probability distribution of [MASK] token can better distinguish the relations between queries and prototypes in meta-learning. In future work, we will expand our method to other tasks, such as few-shot NER and few-shot NER-RE joint extraction.

CRediT authorship contribution statement

Kai He: Conceptualization, Writing – original draft, Software. **Yucheng Huang:** Data curation. **Rui Mao:** Investigation, Validation. **Tieliang Gong:** Supervision. **Chen Li:** Methodology. **Erik Cambria:** 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.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046). This work is also supported by the Key Research and Development Program of Ningxia Hui Nationality Autonomous Region (2022BEG02025); The Key Research and Development Program of Shaanxi Province (2021GXLH-Z-095); The Innovative Research Group of the National Natural Science Foundation of China (61721002); The innovation team from the Ministry of Education (IRT_17R86).

References

Balntas, V., Riba, E., Ponsa, D., & Mikolajczyk, K. (2016). Learning local feature descriptors with triplets and shallow convolutional neural networks. In E. R. H. Richard C. Wilson, & W. A. P. Smith (Eds.), *Proceedings of the British machine vision conference* BMVC, (pp. 119.1–119.11). York, UK: BMVA Press, http://dx.doi.org/10.5244/C.30.119.

Bao, H., He, K., Yin, X., Li, X., Bao, X., Zhang, H., et al. (2021). BERT-based meta-learning approach with looking back for sentiment analysis of literary book reviews. In *International conference on natural language processing and chinese computing* (pp. 235–247). Springer.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. arXiv:2005.14165.

Cambria, E., Liu, Q., Decherchi, S., Xing, F., & Kwok, K. (2022). SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In *LREC* (pp. 3829–3839).

Canese, K., & Weis, S. (2013). PubMed: the bibliographic database. In *The NCBI Handbook, Vol. 2* (p. 1). National Center for Biotechnology Information (US).

Chen, X., Zhang, N., Xie, X., Deng, S., Yao, Y., Tan, C., et al. (2021). KnowPrompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. arXiv:2104.07650.

Dong, B., Yao, Y., Xie, R., Gao, T., Han, X., Liu, Z., et al. (2020). Meta-information guided meta-learning for few-shot relation classification. In *Proceedings of the 28th international conference on computational linguistics* (pp. 1594–1605). Barcelona, Spain (Online): International Committee on Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.coling-main.140, URL: https://aclanthology.org/2020.coling-main.140.

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. arXiv:2012.15723.

Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., et al. (2019). FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 6250–6255). Hong Kong, China: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D19-1649, URL: https://aclanthology.org/D19-1649.

Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., et al. (2020). A deep look into neural ranking models for information retrieval. *Information Processing & Management*, *57*(6), Article 102067.

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

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., et al. (2018). FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4803–4809). Brussels, Belgium: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D18-1514, URL: https://aclanthology.org/D18-1514.

Haviv, A., Berant, J., & Globerson, A. (2021). BERTese: Learning to speak to BERT. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main volume* (pp. 3618–3623). Online: Association for Computational Linguistics, URL: https://aclanthology.org/2021.eacl-main.316.

He, K., Mao, R., Gong, T., Li, C., & Cambria, E. (2022). Meta-based self-training and re-weighting for aspect-based sentiment analysis. *IEEE Transactions on Affective Computing*, (01), 1–13. http://dx.doi.org/10.1109/TAFFC.2022.3202831.

He, K., Yao, L., Zhang, J., Li, Y., Li, C., et al. (2021). Construction of genealogical knowledge graphs from obituaries: Multitask neural network extraction system. *Journal of Medical Internet Research*, *23*(8), Article e25670.

Ji, G., Liu, K., He, S., & Zhao, J. (2017). Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 31*. URL: https://ojs.aaai.org/index.php/AAAI/article/view/10953.

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, URL: https://aclanthology.org/2020.tacl-1.28.

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization. arXiv:1412.6980.

Lan, Y., & Jiang, J. (2021). Modeling transitions of focal entities for conversational knowledge base question answering. 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. 3288–3297). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.acl-long.255, URL: https://aclanthology.org/2021.acl-long.255.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. arXiv:2104.08691.

Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. arXiv:2101.00190.

Lin, Q., Liu, J., Zhang, L., Pan, Y., Hu, X., Xu, F., et al. (2021). Contrastive graph representations for logical formulas embedding. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., et al. (2021). GPT understands, too. arXiv:2103.10385.

Mao, R., & Li, X. (2021). Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35* (pp. 13534–13542).

Mao, R., Li, X., Ge, M., & Cambria, E. (2022). MetaPro: A computational metaphor processing model for text pre-processing. *Information Fusion*, *86–87*, 30–43. http://dx.doi.org/10.1016/j.inffus.2022.06.002.

Mao, R., Liu, Q., He, K., Li, W., & Cambria, E. (2022). The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection. *IEEE Transactions on Affective Computing*, (01), 1–11. http://dx.doi.org/10.1109/TAFFC.2022.3204972.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. http://dx.doi.org/10.48550/ARXIV.1301.3781, URL: https://arxiv.org/abs/1301.3781.

Peng, H., Gao, T., Han, X., Lin, Y., Li, P., Liu, Z., et al. (2020). Learning from context or names? An empirical study on neural relation extraction. In *Proceedings of the 2020 conference on empirical methods in natural language processing* (pp. 3661–3672). virtual conference: Association for Computational Linguistics, URL: https://aclanthology.org/2020.emnlp-main.0.pdf.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018a). Deep contextualized word representations. CoRR, abs/1802.05365. URL: http://arxiv.org/abs/1802.05365. arXiv:1802.05365.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018b). 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). New Orleans, Louisiana: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N18-1202, URL: https://aclanthology.org/N18-1202.

Qu, M., Gao, T., Xhonneux, L.-P., & Tang, J. (2020). Few-shot relation extraction via Bayesian meta-learning on relation graphs. In H. D. III, & A. Singh (Eds.), *Proceedings of machine learning research*: *vol. 119*, *Proceedings of the 37th international conference on machine learning* (pp. 7867–7876). Virtual: PMLR, URL: https://proceedings.mlr.press/v119/qu20a.html.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*, 1–67.

Ren, X., Wu, Z., He, W., Qu, M., Voss, C. R., Ji, H., et al. (2017). CoType: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th international conference on world wide web* WWW '17, (pp. 1015–1024). Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, http://dx.doi.org/10.1145/3038912.3052708.

Roy, A., Ghosal, D., Cambria, E., Majumder, N., Rada, M., & Poria, S. (2022). Improving zero-shot learning baselines with commonsense knowledge. *Cognitive Computation*.

Schick, T., Schmid, H., & Schütze, H. (2020). Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th international conference on computational linguistics* (pp. 5569–5578). Barcelona, Spain (Online): International Committee on Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.coling-main.488, URL: https://aclanthology.org/2020.coling-main.488.

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, URL: https://aclanthology.org/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, URL: https://aclanthology.org/2021.naacl-main.185.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). Autoprompt: Eliciting knowledge from language models using automatically generated prompts. In *Proceedings of the 2020 conference on empirical methods in natural language processing* EMNLP, (pp. 4222–4235). Online: Association for Computational Linguistics, URL: https://aclanthology.org/2020.emnlp-main.346.

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 4080–4090). Red Hook, NY, USA: Curran Associates Inc..

Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., et al. (2021). ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. http://dx.doi.org/10.48550/ARXIV.2107.02137, URL: https://arxiv.org/abs/2107.02137.

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., et al. (2020). ERNIE 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34* (pp. 8968–8975). http://dx.doi.org/10.1609/aaai.v34i05.6428, URL: https://ojs.aaai.org/index.php/AAAI/article/view/6428.

Vu, T., Lester, B., Constant, N., Al-Rfou, R., & Cer, D. (2021). SPoT: Better frozen model adaptation through soft prompt transfer. arXiv:2110.07904.

Wang, Y., Fan, Z., & Rose, C. (2020). Incorporating multimodal information in open-domain web keyphrase extraction. In *Proceedings of the 2020 conference on empirical methods in natural language processing* EMNLP, (pp. 1790–1800). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.emnlp-main.140, URL: https://aclanthology.org/2020.emnlp-main.140.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., et al. (2021). KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, *9*, 176–194.

Wang, J., & Lu, W. (2020). Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 conference on empirical methods in natural language processing* EMNLP, (pp. 1706–1721). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.emnlp-main.133, URL: https://aclanthology.org/2020.emnlp-main.133.

Wen, W., Liu, Y., Ouyang, C., Lin, Q., & Chung, T. (2021). Enhanced prototypical network for few-shot relation extraction. *Information Processing & Management*, *58*(4), Article 102596.

Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Chetvernin, V., et al. (2007). Database resources of the national center for biotechnology information. *Nucleic Acids Research*, *36*(suppl_1), D13–D21.

Xu, J., Kim, S., Song, M., Jeong, M., Kim, D., Kang, J., et al. (2020). Building a PubMed knowledge graph. *Scientific Data*, *7*(1), 1–15.

Yang, S., Zhang, Y., Niu, G., Zhao, Q., & Pu, S. (2021). Entity concept-enhanced few-shot relation extraction. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (Volume 2: Short papers)* (pp. 987–991).