CQARE: Contrastive Question-Answering for Few-shot Relation Extraction with Prompt Tuning

Anonymous ACL submission

Abstract

Prompt tuning with pre-trained language models (PLM) has exhibited outstanding performance by closing the gap between pre-training tasks and various downstream applications, without the need for uninitialized parameters to be introduced. However, prompt tuning requires vast amounts of prompt engineering and predefined label word mapping, which obstructs its implements in practice. Besides, the ample label space makes prompt tuning more arduous and challenging when it comes to relation extraction (RE). To tackle these issues, we propose a Contrastive Question-Answering method with prompt tuning for few-shot RE (CQARE). CQARE carries out a RE task-specific pre-training with four entity-relation-aware pre-training objects, including a prompt pre-training to automatically generate continuous prompts. The proposed pre-training can provide more robust initialization with prompt tuning while maintaining semantic consistency with the proposed PLM. Furthermore, CQARE can effectively avoid label words mapping by reformulating RE as contrastive question answering. The results indicate CQARE raising averaged accuracy of 5.11% on a cross-domain few-shot dataset, demonstrating that robust initialization is crucial for prompt tuning and effective contrastive question answering.

1 Introduction

Relation extraction (RE) is a fundamental task in natural language processing (NLP), aiming to populate knowledge with facts from unstructured text. Many downstream applications rely on extracted relations, such as information retrieval (Guo et al., 2020), question answering (QA) (Lan and Jiang, 2021), and knowledge graph construction (Li et al., 2020). However, most existing RE models are prone to labeled data in practice and face non-negligible challenges due to the variations of domains and languages. Few-shot learning only requires a handful of labeled examples, which has raised more attention in recent research.

For few-shot tasks, GPT-3 (Brown et al., 2020) proves the prominent ability for predictions without any further fine-tuning by fusing manual prompts. Some following studies (Lester et al., 2021; Liu et al., 2021; Vu et al., 2021) explore different methods to tune neural models with prompts and obtain promising results. The main idea behind prompt tuning is reformulating various downstream applications as mask language tasks. The approach closes the gap between the pre-training language model (PLM) and downstream applications without introducing any uninitialized parameters. Benefiting from the above advantages, prompt tuning becomes a promising technology in the low-data regime, without the requirements of sufficient data to train any uninitialized parameters.

Despite the great empirical success, prompt tuning still has two major limitations. The first limitation is that prompt tuning requires extra prompts. Some prompt tuning works (Schick et al., 2020; Schick and Schütze, 2021b,a) make an effort to manually create prompts. However, handcrafting prompts exists three drawbacks: (1) there is not enough annotated data to validate them in a few-shot setting; (2) handcrafting meaningful prompts is brain-draining work, especially designed for an abstract relation with two different entities; (3) 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 automatically prompt designs are proposed (Jiang et al., 2020; Shin et al., 2020; Gao et al., 2021). However, they will suffer a discrete form or isolation problem with employed PLM. A robust initialization for prompt tuning is not paying enough attention in the above studies.

The second limitation is that prompt tuning needs an extra process to map model-predicted words to class labels, named label words mapping.

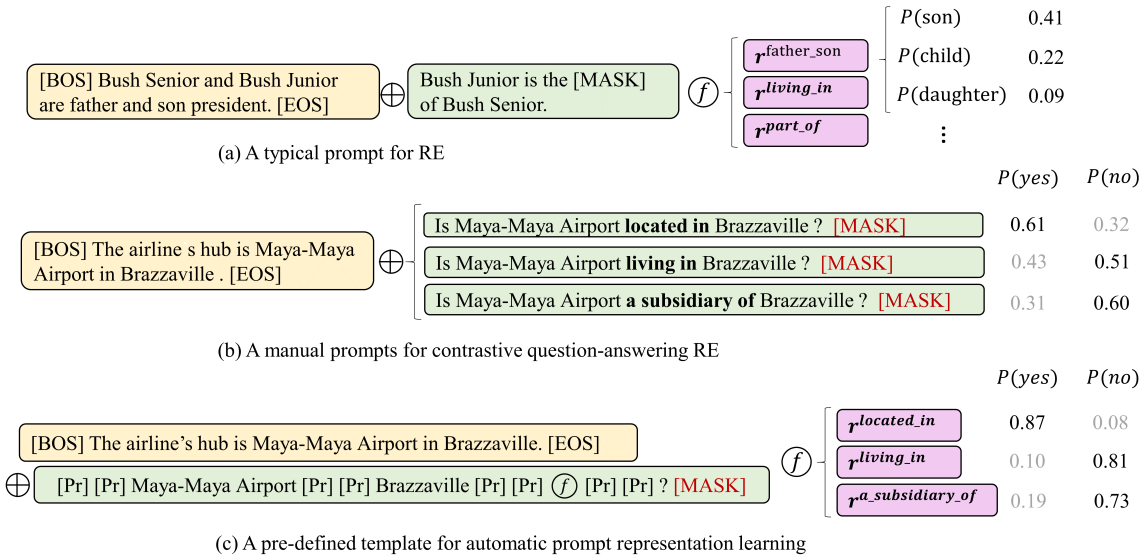

Figure 1: The prompt tuning for few-shot relation extraction. CQARE decodes the last [MASK] into "yes" or "no" answer for predictions. Compared with manual prompts, our automatically generated prompts utilize special marker [Pr] as virtual words instead of natural language. These markers will input the prompt generator to get continuous embeddings. $\oplus$ means the concatenation.

It should notice that the selection of label words depends on empirical attempts entirely. An elaborated label words mapping may also be costly and time-consuming. Most existing prompt tuning studies focus on text classification tasks, where possible categories are no more than five, such as positive or negative in sentiment analysis. When it comes to RE, the label space becomes much larger. For example, the corpus FewRel 2.0 (Gao et al., 2019) contains 124 relations, which makes it highly arduous to construct all label words mapping for each relation with a constant endeavor.

To address the above two limitations, we propose contrastive QA for few-shot RE with prompt tuning (CQARE). CQARE contains four pre-training objects for obtaining an entity-relation-aware PLM, including tasks of mask entity prediction, entity typing, distant supervised RE, and contrastive prompt pre-training. Compared with generating prompts in natural language, CQARE learns the prompt representations in a continual vector space based on an entity-relation orientated string template. Prompt pre-training is included in CQARE, which can learn a more robust initialization for prompt tuning by integrating the contexts, entities, and relation in automatically generated prompts. Moreover, RE is reformulated as a prompt tuning based contrastive QA task, as Figure 1 (b) and (c) shows. Given a context, combined with generated relation-specific prompts, CQARE aims to predict the probabilities of "yes"(positive) or "no"(negative), appearing in a [MASK] position to distinguish whether the two entities share the specific relation. In this way, the multi-relation classification issue can be simplified as a binary classification task. By comparing semantic similarity with positive and negative prototypes, we can exempt cumbersome label word mapping. Our main contributions are threefold:

- **Robust presentations for prompt tuning**. The proposed CQARE contains an entity-relation-aware language model and a non-aggressive prompt generator, enabling joint pre-training to automatically provide prompts in vector space with robust initialization for prompt tuning.
- **Exempt from label words mapping**. We reformulate few-shot RE as a contrastive QA task. By concatenating different prompts as yes-no questions, CQARE can avoid the labor-intensive label word mappings when utilizing prompt tuning in RE tasks.
- **Promising performance**. We expand prompt tuning to few-shot RE tasks. The results demonstrates that CQARE is effective for few-shot RE, raising average accuracy of 5.11% on a cross-domain few-shot dataset. Meanwhile, 18.61% accuracy improvement brought by our pre-training demonstrates that robust initialization is crucial for prompt tuning.

2 Related Work

2.1 Few-Shot Relation Extraction

Generally, few-shot RE can be categorized into two classes. The formal one seeks better presentations through pre-training. Kepler (Wang et al., 2021) integrated knowledge embedding into PLMs by encoding textual entity descriptions and then jointly optimized the knowledge embedding and language modeling objectives. 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 the different predicted methods based on existing PLMs. 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, CQARE simultaneously explores integrating entity and relation knowledge by our particular pre-training tasks and utilizes different approaches for few-shot predictions.

2.2 prompt tuning

GPT-3 (Brown et al., 2020) firstly created manual prompts for different tasks, including translation, QA, and probing tasks. Schick et al. (2020); Schick and Schütze (2021b) were another two early studies that utilized prompts for text classification. They manually constructed prompts, explored semi-supervised solutions for few-shot tasks, and proposed a PET approach to map multi-label words into categories.

Manually constructed appropriate prompts are cumbersome and uncertain. For such reason, some automated prompt creation methods were proposed. Han et al. (2021) applied logic rules to construct prompts, and tried to encode prior knowledge of each class into prompt tuning. AutoPrompt (Shin et al., 2020) explored combining a set of trigger tokens according to a template with the original task inputs to create prompts, and employed a gradient-based search strategy to update them. Gao et al. (2021) utilized separated PLM to generate prompts automatically and dynamically and incorporated demonstrations into context. BERTese (Haviv et al., 2021) adopted a paraphrasing-based approaches to generate prompts. It converted an existing seed prompt to a collection of candidate prompts, and selected ones with the best performance to use. Li and Liang (2021) and Lester et al. (2021) were declared as a lightweight alternative to fine-tuning. These methods froze the parameters of PLM, and only updated a small task-specific vector as prompts. However, these freezing parameters methods became competitive with typical fine-tuning when the used PLM had more than 11 billion parameters (T5 XXL).

KnowPrompt (Chen et al., 2021) was similar to our work, which adapted prompt tuning in RE tasks. The difference was that KnowPrompt focused on injecting entity and relation information into generated prompts. Their prompts were two entity representations concatenated an extra [MASK], which was too brief to provide meaningful information. It should notice that conditional generation based prompt tuning needs enough information for interring in a different context. Different from the above studies, CQARE focused on joint pre-training a prompt generator with PLM for more robust initialization, and explored how to utilize these continuous prompts on few-shot RE tasks.

3 Preliminary

Prompt tuning reformulates downstream applications as mask language tasks. Typically, it needs extra prompts and a predefined label words mapping. A typical prompt is a natural language sentence with a special [MASK] marker. As shown in Figure 1 (a), "Bush Junior is the [MASK] of Bush Senior" is a typical prompts corresponding to the relation type "father_son". With original sentences concatenated prompts as inputs, prompt tuning tries to generate label words $w \in \mathcal{V}_{label}$, where $\mathcal{V}_{label}$ is the predefined label words set. $\mathcal{V}_{label} \in \mathcal{V}$, where $\mathcal{V}$ is the vocabulary of PLM. Next, w will be mapped into a predicted category $c \in \mathcal{C}$ by a label words mapping. The set $\mathcal{C}$ is the label set of the adopted corpus. For example, as shown in Figure 1 (a), if the [MASK] is decoded as "son", this label word will be mapped into the relation "father_son" correctly.

This process proposes a severe challenge in RE tasks by manually creating prompts. For example, "Bush Junior and Bush Senior are father and son presidents" represents a clear relation type "father_son". For such simple instances, manual prompts could be easily designed. However, for

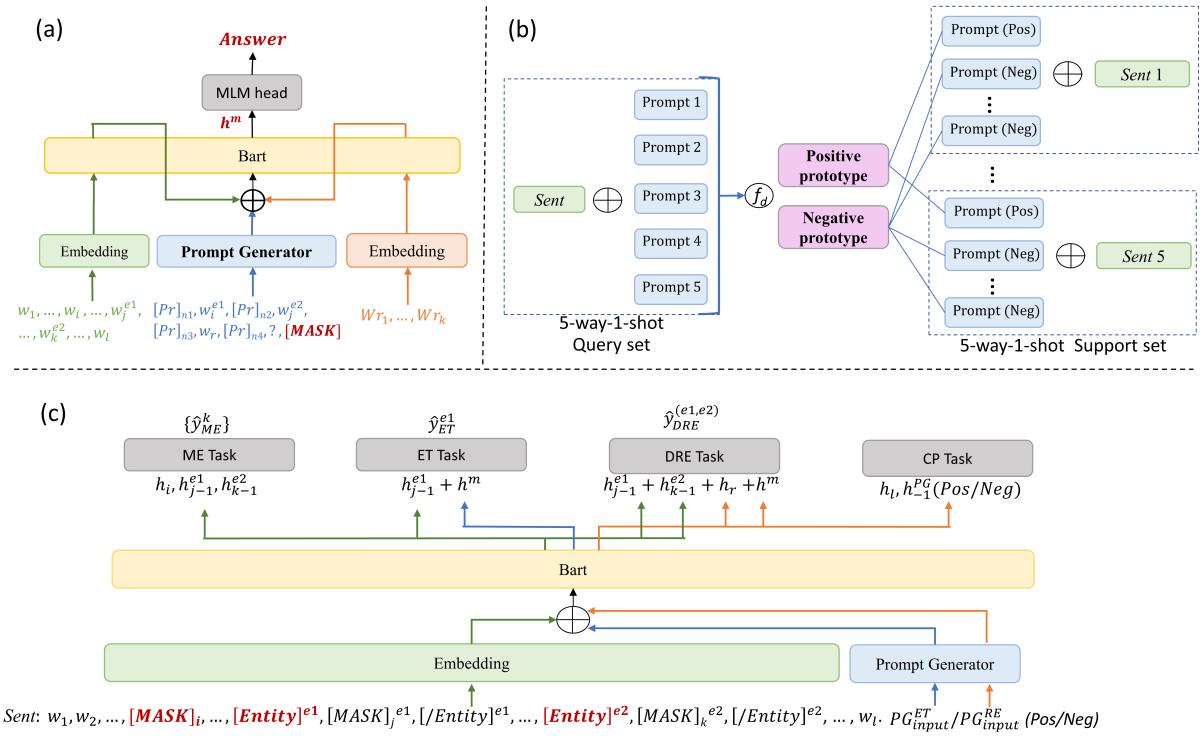

Figure 2: The architecture of CQARE. (a): The module for decoding [MASK] with prompt tuning. CQARE only decodes h_m for obtaining an answer. h^m is the last [MASK]'s hidden state of a prompt generator's input. $w_{r1}, ..., w_k$ is a token sequence of relation name. (b): A example of 5-way-1-shot contrastive QA. (c): Joint pre-training for prompt generator and entity-relation-aware PLM. h_i is i_{th} hidden state of a input sequence. h_l is the representation of $[EOS]$ in $Sent$. $h_{-1}^{PG}(Pos/Neg)$ are positive/negative prompts in Eq. 11. PG_{input}^{ET} and PG_{input}^{RE} are predefined templates for the prompt generator as shown in Eq. 7 and 9.

some complex relations like "place served by transport hub" or "is the primary anatomic site of disease", meaningful and effective prompts are not easy for handcrafts. Besides, the predefined label words for these relations are based on empirical attempts entirely. This process is also costly and unexplainable. It is hard to understand why a label word can outperform other label words with similar semantic meaning. To deal with this issue, CQARE employs several virtual tokens to create a learnable prompt and use a contrastive QA way to avoid the label words mapping, as shown in Figure 1 (b) and (c).

4 Methodology

CQARE consists of a contrastive QA-based few-shot RE with prompt tuning, and a joint pre-training for the prompt generator and PLM. The joint pre-training contains tasks of mask entity prediction (ME), entity typing (ET), distant supervised RE (DRE), and contrastive prompt pre-training (CP). The above tasks make our PLM understand entities and relations better while initializing a prompt generator. Then, the PLM and prompt generator are employed for a prototype-based module. By comparing the instances with positive and negative answer prototypes calculated from support sets, CQARE gets rid of a predefined label words mapping.

4.1 A Contrastive Question-answering with Prompt Tuning

CQARE regards each N-way-K-shot sample as a meta-task $\mathcal{M}$. Each $\mathcal{M}$ will be formulated as $N * K * N$ QA instances. In particular, N relation-specific prompts in $\mathcal{M}$ will be concatenated with each original sentence $Sent$ as different contrastive questions. All inputs of $\mathcal{M}$ are shown as:

$$inputs = \{input_s^n | 1 \leq n \leq N, 1 \leq s \leq N * K\} \quad (1)$$

An input instance $input_s^n$ for CQARE consists of one $Sent$, plusing one prompt with one symbol "?" and one [mask], as shown in the Eq. 2:

$$\begin{aligned} input_s^n = &[BOS] \oplus Sent \oplus [EOS] \\ &\oplus prompt^n \oplus \text{“?”} \oplus [MASK] \oplus [EOS] \end{aligned} \quad (2)$$

where $[BOS]$ is a special marker stands for the start of a sentence, $[EOS]$ for the end of a sentence.

$prompt$ is generated from a non-aggressive prompt generator, which is updated during both the prompt tuning and the pre-training phases, as shown in Figure 2 (a) and (c). With a generated prompt and symbol "?", the representation of the last [MASK] marker will be decoded as answers to these prompt questions, as shown in Figure 1 (a).

Different from standard prompt tuning needs to decode the [MASK] embedding h^m into natural language tokens and map the tokens into the label set with predefined label-word mapping, CQARE directly compares the continuous h^m (answers) in $\mathcal{M}$ to get prediction results. With the symbol "?" and a pre-trained prompts, the answer is prone to positive if the concatenated prompt is correct for the original sentence $Sent$, and negative for a wrong prompt. In such a way, CQARE can simplify few-shot RE as a binary classification QA task.

In particular, CQARE calculates the prototype $Proto_p$ and $Proto_n$ by averaging all the positive answers h_p^m and negative answers h_n^m in the support set of $\mathcal{M}$:

$$Proto_{p/n} = Avg.(h_{p/n}^m) \tag{3}$$

Next, we utilize Euclidean distance d to calculate the probability that the query answers h_i^m are more close to $Proto_p$ or $Proto_n$ in support set:

$$P(\hat{y}_i = p/n) = \frac{exp(-d(h_i^m, Proto_{p/n}))}{\sum_{i \in N} exp(-d(h_i^m, Proto_{p/n})} \tag{4}$$

The final prediction $\hat{y}$ is the most confident positive instance as shown in Eq. 5 and Figure 2 (b).

$$P(\hat{y}) = \arg\max_{i \in N} (P(\hat{y}_i = p)) \tag{5}$$

In such a way, CQARE achieves prompt tuning without listing all possible label words for each relation, avoiding cumbersome labeling and extra unnecessary error. It also simplifies the RE into binary classification, which is proved effective under the few-shot setting in the following experiments. Finally, we use the binary-class cross-entropy loss to train all parameters of CQARE without freezing PLM.

4.2 Joint pre-training for prompt generator and entity-relation-aware PLM

The representations of entity and relation are useful information in various natural language tasks, especially for RE. We design four pre-training tasks to improve entity and relation understanding for PLM, and integrate a prompt generator into the pre-training process for a more robust prompt tuning, as shown in Figure 2 (c).

We first collect general data from a Wikipedia database dump (Attardi, 2015), labeling the entity type with NER tools (spaCy) automatically. The utilized biomedical data with entity information are from (Xu et al., 2020), employed PubMed (Canese and Weis, 2013) as a data resource. Next, we employ distant supervision (Ren et al., 2017; Ji 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), and filter out sentences without any relation.

The first pre-training task is a masked entity (ME) task. Given an input sentence, 10% single tokens and 50% entities are randomly replaced by [MASK], and CQARE tries to decode a single token or a multiple-token entity from each [MASK] marker. The ME pre-training loss is defined as:

$$\mathcal{L}_{ME} = -\sum_{k \in k} \log \prod_{w_i \in words} p(\hat{y}_{ME}^k | w_{\{i \leq k-1\}}) \tag{6}$$

where K is the position set of [MASK] markers, $\hat{y}_{ME}^k$ is the sequence of decoded tokens corresponding k_{th} [MASK] marker.

The second pre-training task is entity typing (ET) with prompts. Firstly, a random initialized prompt generator takes a predefined template, filled with one entity mention e and one sampled entity type $eType$ as inputs PG_{input}^{ET}:

$$\begin{aligned} PG_{input}^{ET} = &[Pr]_{n1} \oplus e \oplus [Pr]_{n2} \oplus eType \\ &\oplus [Pr]_{n3} \oplus \text{“?”} \oplus [MASK] \end{aligned} \tag{7}$$

where n_1, n_2, n_3 is the number of inserted virtual markers [Pr]. The outputs embeddings of prompt generator are the generated prompts PG_h^{ET} for ET task. We denote a positive entity typing prompt by PG_{input}^{ET} if the sampled entity type is correct for the entity mention, otherwise negative prompt. Secondly, CQARE inserts the special marker $[Entity]$ and $[/Entity]$ before and after each annotated entity. We randomly sample equal numbers of positive and negative PG_h^{ET} following the original sentence's input embeddings as inputs of PLM. After encoding by our PLM, the $[Entity]$'s representation h^e and [MASK]'s representations in the last of a prompt h^m will be summed up for the ET task:

$$p(\hat{y}_{ET}^e) \propto Exp(W_e \cdot (h^e + h^m) + b_e) \tag{8}$$

where all w_e and b_e are trainable parameter, y_{ET}^e is the prediction of ET, and cross entropy loss $\mathcal{L}_{ET}$ is calculated for optimization.

Distant supervised RE (DRE) is the third pre-training task. Given a set of labeled entities in one sentence, CQARE combined any two entities as a pair for RE. With the alignment with knowledge base wiki-5m and UMLS, we automatically annotate relations among these entity pairs in a distant supervised way. In particularly, for sampled entity pairs (e^1, e^2), the related prompt inputs are shown as :

$$PG^{RE}_{input} = [Pr]_{n1} \oplus e^1 \oplus [Pr]_{n2} \oplus e^2 \oplus [Pr]_{n3} \oplus relation \oplus [Pr]_{n4} \oplus ? \oplus [MASK] \quad (9)$$

The output embeddings from prompt generator are the generated prompts PG^{RE}_h for RE task. Similar with ET task, PG^{RE}_h is following the raw sentence representation as an input for PLM. After encoding, relation type $\hat{y}^{e_1,e_2}_{DRE}$ is predicted as:

$$p(\hat{y}^{e_1,e_2}_{DRE}|h_r) \propto Exp(W_r \cdot (h^{e1} + h^{e2} + h_r + h^m) + b_r) \quad (10)$$

where h^{e1}, h^{e2}, and h_r are representation of $entity^h$, $entity^t$, and $relation_r$, h^m is the last [MASK]'s representation in Eq. 9. The cross entropy loss $\mathcal{L}_{RE}$ is calculated for optimization.

The last pre-training task is contrastive prompt pre-training (CP). CQARE employs a contrastive triplet loss (Vassileios Balntas and Mikolajczyk, 2016), aiming at learning representations by pulling instances with similar meaning together and pushing different instances apart. In particular, this task takes a raw sentence $Sent$, a positive prompt, and a negative prompt for calculating the loss $\mathcal{L}_{tri}$ as Eq. 11. This loss can ensure the generated prompts are more similar to the corresponding raw sentence, obtaining a "yes" answer with positive prompts.

$$\mathcal{L}_{tri}(a, p, n) = max\{||a_i - p_i||_2 - ||a_i - n_i||_2 + margin, 0\} \quad (11)$$

where a, p, n are the last hidden states of sampled raw sentence h_l, positive and negative prompts h^{PG}_{-1}. Finally, the total loss $\mathcal{L}_{total}$ is formulated as:

$$\mathcal{L}_{total} = \lambda_{ME}\mathcal{L}_{ME} + \lambda_{CP}\mathcal{L}_{CP} + \lambda_{ET}\mathcal{L}_{ET} \cdot I(random) + \lambda_{RE}\mathcal{L}_{RE} \cdot (1 - I(random)) \quad (12)$$

where $\lambda_{ME}, \lambda_{ET}, \lambda_{CP}, \lambda_{RE}$ are the weights of losses, $random \in [0, 1]$ and I(O) is a switching function to choose one task from RE and ET. It is defined as:

$$I(random) = \begin{cases} 1, & if\ random \leq \beta \\ 0, & if\ random > \beta \end{cases} \quad (13)$$

where β is the hyper-parameter for dynamically controlling the ratio of pre-training ET and RE.

5 Experiments

5.1 Experiment Settings

Formulation for N-way-K-shot In this work, we focus on N-way-K-shot RE tasks. It first divides the whole dataset into train, valid, and test sets as usual, without overlapped relation types among them. The training, validation, and test sets are divided into pairs of support sets and query sets. A support set contains N classes randomly sampled from all corpus, and each class has K instances. A query set contains arbitrary instances to be predicted, and the related categories for these instances should be in the corresponding support set. 5-way-1-shot, 5-way-5-shot, 10-way-1-shot, 10-way-5-shot are four common combinations in this setting.

Data We evaluate CQARE on the FewRel (Han et al., 2018) and FewRel 2.0 (Gao et al., 2019), which following the above N-way-K-shot setting. FewRel only focuses on few-shot RE, and its training, validation, and test set all come from wiki data. FewRel 2.0 proposed a few-shot domain adaptation (DA) challenge, which tries to further evaluate across domain abilities of few-shot models. Its validation and test set come from the medical domain, while the train set is still in the general domain.

Evaluation Following the FewRel and FewRel 2.0, we report performances measured by averaged accuracies on the online test set from the official website[1], which contains 10,000 test instances.

Hyper-parameter CQARE utilizes base-BART, with 768 hidden dimensions, and the maximum length of the sentence is 128. Adam optimizer (Kingma and Ba, 2017) is employed with the initial learning rate 2e-5 and batch size 4. n_1, n_2, n_3, n_4 are 1, 3, 3, 1 in Eq .9, respectively. Our experiments utilize 4 A100 GPUs for contrastive question-answering and joint pre-training, and a total of 200G data are employed to pre-train 20 days for our pre-training. The remaining details of CQARE can be obtained from the source code in GitHub[2].

5.2 Main Results

Table 1 shows the accuracy of the test set of DA challenge of FewRel 2.0. We manually create two prompts for each relation in FewRel and FewRel

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

FewRel 2.0 (DA)	5-1	5-5	10-1	10-5	Avg.
Proto-Glove	35.09	49.37	22.98	35.22	35.67
Proto-BERT	40.12	51.50	26.45	36.93	38.75
Pair-BERT	67.41	78.57	54.89	66.85	66.93
Pair-KEPLER	67.23	82.09	54.32	71.01	68.66
Proto-KEPLER	66.41	84.02	51.85	73.60	68.97
CP	79.70	84.90	68.10	79.80	78.12
CQA-B-M	68.74	85.03	55.71	72.11	70.40
CQA-K-M	73.84	89.88	59.52	78.81	75.51
CQA-Ba-M	68.48	84.67	56.50	73.55	70.80
CQA-Ba-C	64.49	76.20	50.09	67.70	64.62
Proto-P	78.12	91.14	63.99	79.59	78.21
CQA-P-M	79.60	91.48	67.69	80.43	79.80
CQA-P-C	**83.39**	**92.97**	**74.32**	**82.23**	**83.23**

Table 1: Accuracy (%) on FewRel 2.0 Domain adaption (DA) challenge. CP (Peng et al., 2020) and KEPLER (Wang et al., 2021) are two recent baseline methods. Proto, Pair, and CQA mean using prototype network (Snell et al., 2017), pair network (Gao et al., 2019) and our contrastive QA; -B, -K, -Ba, and -P mean using BERT, KEPLER, BART, and our PLM, respectively; -M and -C mean using manual or continual prompts generated by CQARE. As CQARE follows the prototype network, CQARE and Proto can be regarded as comparisons between prompt tuning based contrastive QA and typical fine-tuning.

FewRel	5-1	5-5	10-1	10-5	Avg.
Proto-BERT	80.68	89.60	71.48	82.89	81.16
Pair-BERT	88.32	93.22	80.63	87.02	87.30
Proto-KEPLER	88.30	95.94	81.10	92.67	89.50
Pair-KEPLER	90.31	94.28	85.48	90.51	90.14
JAKET	87.40	92.10	78.90	-	-
REGRAB	90.30	94.25	84.09	88.20	89.21
CP	95.10	97.10	**91.10**	94.70	94.50
CQA-B-M	87.83	95.10	82.81	89.11	88.71
CQA-K-M	91.02	96.06	84.15	90.03	90.32
CQA-Ba-M	89.26	94.42	82.50	88.12	88.95
CQA-Ba-C	90.07	94.76	83.30	89.22	89.34
Proto-P	90.18	96.07	86.05	92.90	91.30
CQA-P-M	92.82	96.70	88.39	92.45	92.59
CQA-P-C	**95.32**	**97.84**	90.08	**95.96**	**94.80**

Table 2: Accuracy (%) on FewRel dataset. JAKET (Yu et al., 2020), REGRAB (Qu et al., 2020), and CP (Peng et al., 2020) are three recent baseline studies. JAKET does not report the 10-way-5-shot result. Acronym is the same with Table 1.

2.0 datasets to compare with the continual prompts generated by CQARE. Proto and Pair are prototype network (Snell et al., 2017) and pair network (Gao et al., 2019). As CQARE is based on the prototype network, Proto can be compared with CQARE for analyzing the reasons of improvements. By keeping the same PLM, the results in Table 1 show prompt tuning has a promising few-shot ability. For example, CQARE-B-M raises average accuracy of 31.65% and 3.47% compared with Proto-BERT and Pair-BERT, and raises 6.54% and 6.85% compared with Proto-KEPLER and Pair-KEPLER. Similar conclusions can also be found in Table 2. CQARE-B-M raises of 7.55% compared with Proto-BERT.

Besides, the results confirm that the different PLMs have significant effects on few-shot RE, especially for DA tasks. The performance gap between Proto-Glove and Proto-KEPLER reaches 33.33%. These gaps are even more apparent when compared with CQARE-P. By comparing -Ba (original BART model) and -P (our PLM) in both Table 1 and Table 2, it confirms proposed entity-relation-aware pre-training obtains 9.00% and 3.84% average accuracy improvements, by using manual prompts. These improvements become more significant by achieving 18.61% in FewRel 2.0 and 5.46% in FewRel when using continual prompts.

Comparing -M and -C, the related results represent the continual and manual prompts' effects. Without pre-training, the continual prompts are only slightly better than manual prompts when combined with original BART in FewRel. CQARE-Ba-C only raise average accuracy of 0.39% compared with CQARE-Ba-M in Table 2. It is even worse than manual prompts in the cross-domain FewRel 2.0 (see CQARE-Ba-C versus CQARE-Ba-M in Table 1). However, the advantages of continual prompts are represented when the prompt generator is joint pre-training with employed PLM. With robust initialization (-P), continual prompts outperform manual prompts by 3.43% in Table 1 and 2.21% in Table 2 (CQARE-P-M versus CQARE-P-C).

6 Discussion

Why our continual prompt outperforms manual or discrete prompt ? First of all, the proposed prompt generator contains a certain amount of pre-training parameters. These parameters can enable CQARE to output different prompts for different contexts, even with the same relation. In particular, a triple (entity1, relationship, entity2) and its different context both decide the prompt representation after encoding by PLM. It is significantly different from typical prompt tuning studies, which usually utilize the unchanged prompt for each class. Besides, all generated prompts from CQARE take virtual markers [Pr] as parts of inputs, which have no any specific semantics. In such condition, these virtual markers can be trained by the context infor-

5-1	$[n_1,n_2,n_3,n_4]$	Pat_1	Pat_2	Pat_3
CQARE -Ba-C	[1, 1, 1, 1]	64.57	63.72	65.55
	[2, 2, 2, 2]	67.45	62.92	66.25
	[1, 3, 3, 1]	67.57	62.32	65.53
	[3, 3, 3, 3]	63.32	62.52	65.37
CQARE -P-C	[1, 1, 1, 1]	84.96	81.42	83.44
	[2, 2, 2, 2]	85.33	82.77	84.01
	[1, 3, 3, 1]	87.62	81.73	84.63
	[3, 3, 3, 3]	87.58	83.95	86.57
CQARE -Ba-M	-	M_1	M_2	
	-	69.75	62.13	

Table 3: The effects for different patterns of prompts. n_1, n_2, n_3, n_4 are control parameters for [Pr] in each position. The reported accuracy is the results on the 5-way-1-shot validation set of Fewrel 2.0. The Pat_1 is $[P_1, e^h, P_2, e^t, P_3, r, P_4, M]$, where P_i means $[Pr]n_i$, e^h means $entity^h$, e^t means $entity^t$ as Eq. 9 shown. Pat_2 is $[P_1, r, P_2, e^h, P_3, e^t, P_4, M]$, and Pat_3 is $[P_1, e^h, P_2, r, P_3, e^t, P_4, M]$. M_1 and M_2 are two sets of manual prompts finished by two isolated annotators.

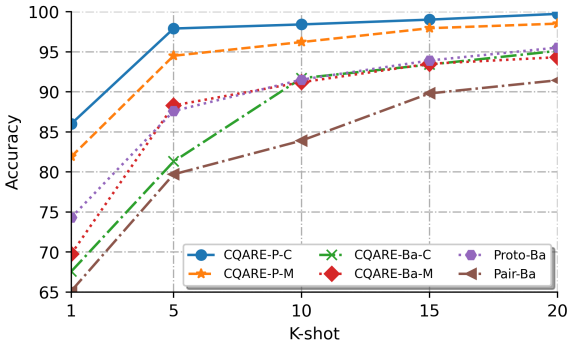

Figure 3: The comparison of combinations of continual prompts, manual prompts, original BART and our PLM with the raising numbers of training instances. The reported accuracy is on the validation set of 5-way-K-shot of Fewrel 2.0. Acronym is the same with Table 1.

mation to obtain a more unbiased representation.

It should notice that the generation of natural language prompts may lose some information. PLM usually carries out a LogSoftMax operation for the continual representation and takes the most confident index to output natural language tokens. These tokens will be further replaced by other PLMs in downstream tasks with new representations corresponding to generated tokens rather than the original continual representation. To this end, the discrete prompts may be confined to isolated natural language tokens, while the original continual representation is in a specific context. Such inconsistencies may cause the neural network hard to achieve global optimum when utilizing prompt tuning for predictions.

Does the pattern of prompts matter ? Liu et al. (2021) presents an example that using different manual prompts on the same instance results in a 19.79% P@1 measure gap. Considering this problem, we explore the effects of a continual prompt with different patterns, namely different numbers and positions for inserted [Pr] markers. As shown in Eq. 9, we try different hyperparameters n_1, n_2, n_3, n_4 as [1,1,1,1], [2,2,2,2], [1, 3, 3, 1], and [3,3,3,3], respectively. We also try to replace Eq. 9 as other patterns, as shown in Table 3. The results indicate that the continual prompts are also influenced by their patterns. When the accuracy gap reaches 7.62% for the two sets of manual prompts, different patterns of continual prompts also have a maximum 6.20% difference. Besides, the patterns play a more influenced role compared with the numbers of [Pr] in each position.

How does the number of data effects prompt tuning ? Considering the advantages of prompt tuning for few-shot tasks, we compared different methods with CQARE under increasing data quantities. As Figure 3 indicates, all methods benefit from more shots, while CQARE outperforms than prototype and pair network (Gao et al., 2019) more obviously when the data number is no more than 10. When data numbers increase to 10-shot and 15-shot, it has fewer effects for CQARE, while prototype and pair network still keep growing. This phenomenon confirms that prompt tuning has the promising ability in the low-data regime. Besides, our entity-relation-aware pre-training also can alleviate the data dependence to some extent.

7 Conclusion and Future Work

This paper proposed CQARE, which expands prompt tuning to few-shot RE tasks. CQARE utilized continual prompts automatically generated from a pre-trained generator. By reformulating few-shot RE as a contrastive QA, CQARE eliminates the labor-intensive label words mapping in the task with large label spaces. The results demonstrate that robust initialization is crucial for prompt tuning and contrastive question answering is effective. Intuitively, the study can be further explored by (1) overcoming the instability associated with patterns of continual prompts; (2) designing better prompt pre-training tasks to integrate PLM and prompt tuning.

References

Giuseppe Attardi. 2015. Wikiextractor. https://github.com/attardi/wikiextractor.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Kathi Canese and Sarah Weis. 2013. Pubmed: the bibliographic database. *The NCBI Handbook*, 2:1.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction.

Bowen Dong, Yuan Yao, Ruobing Xie, Tianyu Gao, Xu Han, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2020. Meta-information guided meta-learning for few-shot relation classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1594–1605, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 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)*, pages 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 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*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Adi Haviv, Jonathan Berant, and Amir Globerson. 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*, pages 3618–3623, Online. Association for Computational Linguistics.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

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

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Yunshi Lan and Jing Jiang. 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)*, pages 3288–3297.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Linfeng Li, Peng Wang, Jun Yan, Yao Wang, Simin Li, Jinpeng Jiang, Zhe Sun, Buzhou Tang, Tsung-Hui Chang, Shenghui Wang, et al. 2020. Real-world data medical knowledge graph: construction and applications. *Artificial intelligence in medicine*, 103:101817.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 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*, pages 3661–3672, virtual conference. Association for Computational Linguistics.

Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via Bayesian meta-learning on relation graphs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7867–7876. PMLR.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, Tarek F. Abdelzaher, and Jiawei Han. 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, page 1015–1024, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 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*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 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*, pages 2339–2352, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 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)*, pages 4222–4235, Online. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 4080–4090, Red Hook, NY, USA. Curran Associates Inc.

Daniel Ponsa Vassileios Balntas, Edgar Riba and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 119.1–119.11. BMVA Press.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

David L Wheeler, Tanya Barrett, Dennis A Benson, Stephen H Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M Church, Michael DiCuccio, Ron Edgar, Scott Federhen, et al. 2007. Database resources of the national center for biotechnology information. *Nucleic acids research*, 36(suppl_1):D13–D21.

Jian Xu, Sunkyu Kim, Min Song, Minbyul Jeong, Donghyeon Kim, Jaewoo Kang, Justin F Rousseau, Xin Li, Weijia Xu, Vetle I Torvik, et al. 2020. Building a pubmed knowledge graph. *Scientific data*, 7(1):1–15.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2020. Jaket: Joint pre-training of knowledge graph and language understanding.