SPE: Symmetrical Prompt Enhancement for Factual Knowledge Retrieval

Anonymous ACL submission

Abstract

Pretrained language models (PLMs) have been shown to accumulate factual knowledge from their unsupervised pretraining procedures (Petroni et al., 2019). Prompting is an effective way to query such knowledge from PLMs. Recently, continuous prompt methods have been shown to have a larger potential than discrete prompt methods in generating effective queries (Liu et al., 2021a). However, these methods do not consider symmetry of the task. In this work, we propose Symmetrical Prompt Enhancement (SPE), a continuous prompt-based method for fact retrieval that leverages the symmetry of the task. Our results on LAMA, a popular fact retrieval dataset, show significant improvement of SPE over previous prompt methods.

1 Introduction

Prompt-based learning proposes to formulate different NLP tasks into language modeling problems (Schick and Schütze, 2021). It is a novel paradigm that effectively uses Pretrained Language Models (PLMs) (Liu et al., 2021a), and achieves comparable or better performance than fine-tuning (Lester et al., 2021). Prompt-based learning has also been used for the task of fact retrieval from PLMs. In this task, the goal is to predict the (masked) object of factual tuples of type (subject, relation, object). Prompt-based methods assume that PLMs gather and store factual knowledge during their pretraining, and cloze-style prompts can retrieve this knowledge (Petroni et al., 2019). The prompts are either handcrafted (Petroni et al., 2019; Bouraoui et al., 2020) or automatically generated (Shin et al., 2020; Haviv et al., 2021). For example, to retrieve the knowledge about geographic location of *Luxembourg* in the PLMs, a `prompt` can be formed by filling *Luxembourg* in the first blank of the following `template`: "____ is located in ____.". An effective prompt will query the PLM to output *Europe* as the most likely prediction for the second blank. Such methods are promising but brittle. Minor changes in the template can lead to significant difference in the performance (Jiang et al., 2020). Recent works has shown that continuous prompts obtained via gradient-based learning, are more effective and robust than discrete prompts since there are less restrictions on the search space (Liu et al., 2021b; Qin and Eisner, 2021; Zhong et al., 2021; Liu et al., 2021a).

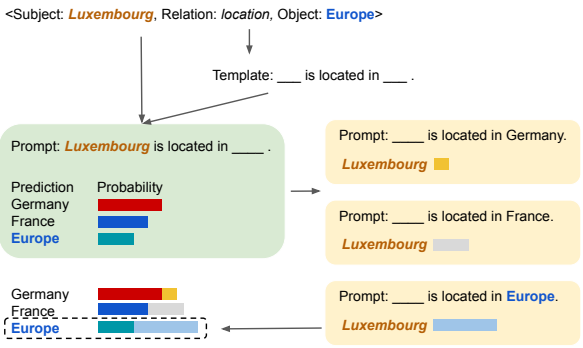

Figure 1: Example of fact retrieval: Given a subject and relation, predict the object. SPE uses a fixed template to generate a prompt for predicting object given subject (green box) as well as several symmetrical prompts for predicting the subject given object candidates (yellow boxes). The final prediction is obtained using the likelihoods of the object candidates and of the given subject as obtained using the symmetrical prompts. Bars represent probabilities from BERT. Note that SPE is a continuous prompt-based method. We use natural language prompts and template here for illustration.

While existing work focuses on the design of the prompts, they do not leverage the symmetry inherent in the task's definition. For example, while *Luxembourg* is located in *Europe*, *Europe* is the region that contains *Luxembourg*. Similar principle is applied in other NLP tasks (Crawford et al., 1996; Kiddon and Domingos, 2015; He et al., 2017; Tanchip et al., 2020).

In this work, we propose *Symmetrical Prompt*

Enhancement (SPE), a continuous prompt learning method that incorporates the above mentioned symmetry of the task. Specifically, in addition to using a prompt to predict the object given the subject, SPE also uses an additional symmetrical prompt to predict the subject given the object. Using the first prompt (see green box in Fig. 1), SPE obtains a few high-probability candidates for the object like *Germany*, *France*, and *Europe*. Thereafter, for each object candidate, it generates a symmetrical prompt (shown in yellow boxes), and obtains the likelihood of the subject, *Luxembourg*. Finally, SPE reranks the object candidates by joint likelihood of both the candidates as well as the subject (given the candidates). In the running example illustrated in Fig. 1, we can see that even though the correct answer, Europe, was not the most likely output in the green box, SPE's symmetrical prompting resulted in its (joint) likelihood being the highest. Our experiments on the fact retrieval dataset, LAMA (Petroni et al., 2019), shows SPE achieves significant improvement over previous prompt approaches.

2 Symmetrical Prompt Enhancement

The goal of fact retrieval via prompt generation is to output object $\mathcal{O}$ for given subject $\mathcal{I}$ and relation $\mathcal{R}$ by constructing a prompt $\mathcal{P}$. Most methods operate by assuming a template $\mathcal{T}$, and generating the prompt $\mathcal{P}$ from $\mathcal{T}$, $\mathcal{I}$ and $\mathcal{R}$. Fig. 1 shows an example of Subject (*Luxembourg*), Relation (*location*), Object (*Europe*), Template (____ *is located in* ____.), and Prompt (*Luxembourg is located in* ____.). Note that the figure shows a natural language template and prompts for readability. However, for continuous prompt methods like ours, the template is a sequences of vectors like $[V]_1...[V]_n$ ____ $[V]_{n+1}...[V]_{n+m}$ ____ $[V]_{n+m+1}...[V]_{n+m+k}$, $\forall [V]_i \in \mathbb{R}^d$. The prompt, $\mathcal{P}$, is typically generated by learning these vectors from the training data and filling the (representation of) $\mathcal{I}$ in the first blank. Note that prompts are relation-specific ($\mathcal{P}^{\mathcal{R}}$) but here we refer to them as $\mathcal{P}$ for simplicity. The model's prediction, $\hat{\mathcal{O}}$, is the most likely object candidate for the remaining blank in the prompt as determined by the PLM. Mathematically,

$$\mathcal{P} = \mathrm{PGen}_\theta(\mathcal{T}, \mathcal{I}) \quad (1)$$

$$\hat{\mathcal{O}} = \arg\max_{v \in \mathcal{V}} \mathrm{P}_{\mathrm{PLM}}(\mathrm{blank} = v | \mathcal{P}), \quad (2)$$

where PGen_θ is the prompt generator parameterized by θ and $\mathcal{V}$ is the vocabulary of the PLM.

Our proposed approach, *Symmetrical Prompt Enhancement* (SPE), leverages the inherent symmetry of the task. Specifically, in addition to learning the original prompt $\mathcal{P}_{orig}$ for predicting the object given the subject, SPE also generates several symmetrical prompts, $\mathcal{P}_{sym}$, for predicting the subject given the object. Like $\mathcal{P}$, $\mathcal{P}_{sym}$ is also generated from $\mathcal{T}$ except that this time the first blank is filled by the (representation of) $\mathcal{O}$.

$$\mathcal{P}_{\mathrm{orig}} = \mathrm{PGen}_\theta(\mathcal{T}, \mathcal{I}) \quad (3)$$

$$\mathcal{P}_{\mathrm{sym}} = \mathrm{PGen}_\theta(\mathcal{T}, \mathcal{O}) \quad (4)$$

$$\hat{\mathcal{O}} = \arg\max_{v \in \mathcal{V}} \mathrm{P}_{\mathrm{PLM}}(\mathrm{blank} = v | \mathcal{P}_{\mathrm{orig}}) \quad (5)$$

$$\hat{\mathcal{I}} = \arg\max_{v \in \mathcal{V}} \mathrm{P}_{\mathrm{PLM}}(\mathrm{blank} = v | \mathcal{P}_{\mathrm{sym}}). \quad (6)$$

The model is trained by optimizing a linear combination of the cross-entropy objectives of predicting the object $\mathcal{O}$ and the subject $\mathcal{I}$:

$$\min_\theta \mathcal{L}_{CE}(\hat{\mathcal{O}}, \mathcal{O} | \mathcal{P}_{orig}) + \lambda \mathcal{L}_{CE}(\hat{\mathcal{I}}, \mathcal{I} | \mathcal{P}_{sym}), \quad (7)$$

where λ is a hyperparameter.

During inference, SPE selects top K predictions $\mathcal{C}^{\mathrm{K}}$ using the original prompt and $\mathcal{I}$, and uses them as candidates to generate symmetrical prompts $\mathcal{P}^{\mathrm{k}}_{sym}$.

$$\mathcal{C}^{\mathrm{K}} = \mathrm{TopK}_{v \in \mathcal{V}} \mathrm{P}(v | \mathcal{I}, \mathcal{P}_{\mathrm{orig}}) \quad (8)$$

$$\mathcal{P}^{\mathrm{k}}_{\mathrm{sym}} = \mathrm{PGen}_\theta(\mathcal{T}, c^{\mathrm{k}}), \quad \forall c^{\mathrm{k}} \in \mathcal{C}^{\mathrm{K}}. \quad (9)$$

Finally, the model's prediction $\hat{\mathcal{O}}$ is:

$$\begin{aligned} \hat{\mathcal{O}} = \arg\max_{c^{\mathrm{k}} \in \mathcal{C}^{\mathrm{K}}} & \log \mathrm{P}_{PLM}(c^{\mathrm{k}} | \mathcal{P}_{\mathrm{orig}}) \\ & + \lambda \log \mathrm{P}_{PLM}(\mathcal{I} | \mathcal{P}^{\mathrm{k}}_{\mathrm{sym}}). \end{aligned} \quad (10)$$

In practice, $\mathcal{L}$ and $\mathcal{P}_{PLM}$ are normalized by input length to account for inputs with multiple tokens.

3 Experimental Setup

We conduct experiments on the fact retrieval part of LAMA dataset (Petroni et al., 2019), which consists of fact triples with single-token objects from 41 relations in Wikidata (Vrandečić and Krötzsch, 2014). We use the training set extended by Shin et al. (2020). We choose masked language models BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as PLMs, which are fixed during training to serve as static knowledge bases. For implementation, we use PLMs in Huggingface library of Transformers (Wolf et al., 2020). Following Liu

Model	BERT-base			BERT-large			RoBERTa-base		
	P@1	P@10	MRR	P@1	P@10	MRR	P@1	P@10	MRR
Manual	31.1	59.5	40.3	28.9	57.7	38.7	-	-	-
LPAQA	34.1	62.0	43.6	39.4	67.4	49.1	1.2	9.1	4.2
AutoPrompt	43.3	73.9	53.9	-	-	-	40.0	68.3	49.9
OptiPrompt (manual)	48.6	79.0	58.9	50.6	79.2	60.7	-	-	-
SoftPrompt (mined)	48.8	79.6	59.4	51.0	81.4	59.6	40.6	75.5	53.0
P-tuning	48.2	78.1	58.6	49.9	80.6	60.6	43.5	73.9	53.8
SPE	**50.3**	**80.5**	**60.9**	**52.3**	**82.2**	**62.9**	**46.4**	**75.5**	**56.1**

Table 1: Result on LAMA. Our approach, SPE, outperforms both discrete prompt approaches: Manual (Petroni et al., 2019), LPAQA(Jiang et al., 2020), and AutoPrompt (Shin et al., 2020); and continuous prompt methods: Optiprompt (Zhong et al., 2021), Softprompt (Qin and Eisner, 2021) and P-tuning (Liu et al., 2021b).

Model		P@1	P@10	MRR
P-tuning		48.2	78.1	58.6
SPE	K=1	48.7	79.9	59.5
	K=5	49.9	79.9	60.5
	K=10	49.9	79.9	60.7
	K=15	**50.3**	**80.5**	**60.9**

Table 2: Effect of varying size of candidate pool on SPE's performance. SPE outperforms P-tuning even without reranking (K=1). A larger candidate pool helps the model even further.

et al. (2021b) we use the following generic format for template, $\mathcal{T}$: $[V]_1$ $[V]_2$ $[V]_3$ ____ $[V]_4$ $[V]_5$ $[V]_6$ ____ $[V]_7$ $[V]_8$ $[V]_9$ $\forall [V]_i \in \mathbb{R}^d$. We also use their BiLSTM (Graves et al., 2013) with multilayer perceptron (MLP) architecture to setup PGen.

For $\mathcal{I}$ with multiple tokens, we mask them one token at a time to generate $\mathcal{P}_{\mathrm{sym}}$, and use the average of pseudo likelihoods from all $\mathcal{P}_{\mathrm{sym}}$s to represent $\mathrm{P}_{PLM}(\mathcal{I}|\mathcal{P}_{\mathrm{sym}})$. In practice, we find that masking one token at a time is better than masking the entire phrase at once, and averaging the pseudo-likelihood has better performance. The training batch size is 8. We set K to be 15 during inference, and λ to be 0.8 according to preliminary results. The results are evaluated by accuracy at top 1 (P@1) and top 10 (P@10) predictions, and Mean Reciprocal Rank (MRR) as in Qin and Eisner (2021). See Appendix A for more setup details.

4 Results

We compare our results with both discrete and continuous prompt methods. Discrete prompt methods include prompts from manually designed templates (Petroni et al., 2019); LPAQA (Jiang et al., 2020), which uses text mining based prompts; and AutoPrompt (Shin et al., 2020), which uses discrete lexicalized trigger tokens for prompt generation. Continuous prompt methods include P-tuning (Liu et al., 2021b), which uses a neural network to generate prompt tokens; OptiPrompt (Zhong et al., 2021), which uses manually initialized continuous prompts; and SoftPrompt (Qin and Eisner, 2021), which ensembles multiple prompts initialized with mined templates.

Our results in Table 1 show that SPE outperforms all previous methods. Note that, unlike OptiPrompt and SoftPrompt, we do not make use of manual templates as initialization. Nevertheless, SPE outperforms them indicating that the it generates prompts of higher qualities even without manual efforts. For the rest of our experiments, we consider P-tuning as our primary baseline since it is the best performing model that is directly comparable to SPE.

Table 2 shows how the performance of SPE varies with the size of the candidate pool. Comparing the first two rows we can see that even with a single candidate (K=1), SPE outperforms our primary baseline, P-tuning. Increasing the size of the candidate pool further improves the performance by allowing the model to conduct bidirectional filtering. However, expanding the candidate pool has trade-off between performance and memory usage. Table 3 shows some qualitative examples of top 5 predictions for a given subject-relation pair from P-tuning (top half of each row) and SPE (bottom half of each row). The correct answers are underlined, and their ranks in the predicted list are shown in the

Rel	Subject	Top 5 Predictions (Prob. High $\longrightarrow$ Low): Top - PT, Bottom - SPE					Rank
P101	Richard Wagner	music	history	psychology	opera	linguistics	4
		opera	music	philosophy	aesthetics	art	1
P108	Spike Milligan	Microsoft	IBM	Google	BBC	ESPN	4
		BBC	Microsoft	CBS	ESPN	Google	1
P364	Baaz	Turkish	English	French	Arabic	Persian	41
		Hindi	Urdu	Punjabi	Bengali	Persian	1
P27	Rubens Barrichello	Belgium	France	Italy	Spain	Germany	15
		Brazil	Spain	Argentina	Portugal	Uruguay	1
P127	Nismo	Google	Nokia	Iceland	Intel	Microsoft	14
		Toyota	Nissan	Honda	Mitsubishi	Volkswagen	2
P30	Marshall Islands	Antarctica	Asia	Africa	Oceania	Europe	4
		Asia	Oceania	Africa	Antarctica	Europe	2

Table 3: Comparison between P-tuning (referred as PT) and SPE for the following relations: P101 (field of work), P108 (employer), P364 (original language of film or TV show), P27 (country of citizenship), P127 (owned by), and P30 (continent). Correct answers are underlined and the last column represents the rank of correct answers.

right-most column. We make several observations from this table.

First, we can see that, in general, the rank of the correct answer is lower in SPE's list than in P-tuning's list. For instance, consider the first row with subject as *Richard Wagner*, a German composer, and relation as P101 (*field of work*). SPE correctly predicts *opera* as the top-ranked object for this example while it appears at the fourth position in P-tuning's list. Similarly, SPE correctly predicts *BBC* as the *employer* of *Spike Milligan*, and *Hindi* as the *original language* of the Indian thriller, *Bazz*.

Second, SPE also correctly identifies the *country of citizenship* for *Rubens Barrichello* as *Brazil*. Identifying objects for relations like *country of citizenship* for individuals are challenging because their personal descriptions appeared in the pretraining corpus of PLMs might contain mentions of several places he/she has worked or lived or received education in. This might create confusion for PLMs. For example, the Wikipedia page of the famous Brazilian Formula One player, *Rubens Barrichello*, mentions a handful of other countries where he participated in competitions.

Third, SPE, in general, brings a notable improvement in the ranked lists, even if the correct answer is not the topmost prediction. For example, *Nismo* is more likely to be *owned by* a Japanese vehicle company than an Internet firm. SPE's top predictions include *Toyota*, *Nissan* (the correct answer), and *Honda*, while P-tuning's top predictions include *Google*, *Nokia*, and *Iceland*. Similarly, SPE's top predictions for original language of Indian thriller *Bazz* include several Indian languages (with the correct answer as the topmost prediction) while P-tuning's top predictions contain European and Middle Eastern languages.

Fourth, for relations with close-set answers (e.g. P30 *continent* of *Marshall Islands*), the task of fact retrieval reduces to a classification problem with fixed number of labels. Prompt based models, in general, are observed to be affected by label imbalance in the training set (Zhong et al., 2021). For example, in our dataset, the majority class for continents is *Antartica* (95.6% of continent-type objects) while *Oceania*, only occurs in 0.4% of the continent-type objects. P-tuning is probably affected by this imbalance and outputs the majority label, *Antartica*, as the continent that contains *Marshall Islands* while *Oceania*, the correct answer, appears at rank 4. SPE is less affected by the abundance of the majority class and missing labels and outputs *Oceania* at the second position.

5 Conclusion

Prompt-based learning is an effective way of knowledge retrieval from PLMs. In this work, we introduce Symmetrical Prompt Enhancement (SPE) that utilizes the inherent symmetry of the task to better improve fact retrieval. Our experiments show that SPE outperforms existing SOTA methods. It also demonstrates potential in alleviating the problem of label imbalance in prompting.

References

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *AAAI*.

James M. Crawford, Matthew L. Ginsberg, Eugene M. Luks, and Amitabha Roy. 1996. Symmetry-breaking predicates for search problems. In *KR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *AISTATS*.

Alex Graves, Abdel rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

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.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1766–1776, Vancouver, Canada. Association for Computational Linguistics.

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.

Chloé Kiddon and Pedro M. Domingos. 2015. Symmetry-based semantic parsing.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ArXiv*, abs/2107.13586.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *ArXiv*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. 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.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with 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.

Chelsea Tanchip, Lei Yu, Aotao Xu, and Yang Xu. 2020. Inferring symmetry in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2877–2886, Online. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning

to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

A Additional Setup Details

The prompt generator consists of a two-layer BiLSTM and a two-layer MLP on top of it. The MLP uses ReLU (Glorot et al., 2011) as the activation function. The hidden size of LSTM and dimension of d are 768 for BERT-base-cased and RoBERTa-base, and 1024 for BERT-large-cased.