KLAPrompt: Infusing Semantic Knowledge into Pre-trained Language Models by Long-answer Prompt Learning

Zuotong Xie[1], Kai Ouyang[1], Xiangjin Xie[1], Hai-Tao Zheng[1,3*] , Wenqiang Liu[2], Dongxiao Huang[2], Bei Wu[2]
[1]Shenzhen International Graduate School, Tsinghua University, China
[2]Interactive Entertainment Group Tencent Inc, China
[3]Pengcheng Laboratory, Shenzhen, China

***Abstract*—Pre-trained language models (PLMs) with external knowledge have demonstrated their remarkable performance on a variety of downstream natural language processing tasks. The typical methods of integrating knowledge into PLMs are designing different pre-training tasks and training from scratch, which requires high-end hardware, massive storage resources, and computing time. Prompt learning is an effective approach to tune PLMs for specific tasks, and it can also be used to infuse knowledge. However, most prompt learning methods accept one token as the answer instead of multiple tokens. To tackle this problem, we propose the long-answer prompt learning method (KLAPrompt) to incorporate semantic knowledge in Xinhua Dictionary into pre-trained language models. The proposed method splits the whole answer space into several answer subspaces according to the token's position in the long answer. Extensive experimental results on five datasets demonstrate the effectiveness of our approach.**

***Index Terms*—semantic knowledge, pre-trained language model, prompt learning**

I. Introduction

In recent years, pre-trained language models (PLMs) with external semantic knowledge have shown excellent performance on many natural language processing (NLP) tasks, including named entity recognition [1]–[4], relation extraction [5]–[8], and machine translation [9]–[12]. However, traditional approaches of introducing knowledge are mostly training from scratch, which is time-consuming and computationally expensive, making it infeasible for most users. Recently, prompt learning has achieved promising results for certain few-shot classification tasks [13]–[16], and it can also be used to integrate knowledge. Xinhua Dictionary [17], the most authoritative and influential modern Chinese dictionary, contains massive and comprehensive content such as word-forms, pronunciations, precise definitions, and rich examples. As shown in Table I, the "sense" is composed of a long string of tokens, but the typical methods of prompt learning accept one token as the answer.

To address this challenge, we propose the long-answer prompt learning method (KLAPrompt) and collect a word sense prediction dataset (WSP) based on Xinhua Dictionary to introduce fine-grained semantic knowledge. Firstly, instead of considering the long answer as a whole, we split the answer space into several answer subspaces according to the token's position in the long answer. For instance, the answer subspaces of "order" in Table I are { "the", "a" }, { "way", "$state$", "$request$" }, { "in", "$that$", "for" }, ..., { "for"}. Then, we train pre-trained language models on WSP dataset to predict the sense, and each word of the sense will be predicted independently.

Word	Sense	Phrase
order	the way in which people or things are placed or arranged in relation to each other	in alphabetical order
		in chronological order
		in descending/ascending order
	the state that exists when people obey laws, rules or authority	keep the class in good order
		maintain order in the capital
		restore public order
	a request for food or drinks in a restaurant; the food or drinks that you ask for	May I take your order?
		an order for steak and fries
		a side order

TABLE I: An example of the word, senses, and phrases in the dictionary.

We conduct comprehensive experiments on five public NLP datasets. Experimental results demonstrate that pre-trained language models gain superior performances on the strength of the semantic knowledge in Xinhua Dictionary. And empirical studies also verify the effectiveness of the KLAPrompt approach in integrating semantic knowledge.

In a nutshell, the main contributions of our work are as follows:

1) We introduce more abundant and fine-grained semantic knowledge in Xinhua Dictionary into the pre-trained language models, enhancing the model's ability to understand Chinese word semantics.

* Corresponding author. (Email: zheng.haitao@sz.tsinghua.edu.cn)
DOI reference number: 10.18293/SEKE2023-200

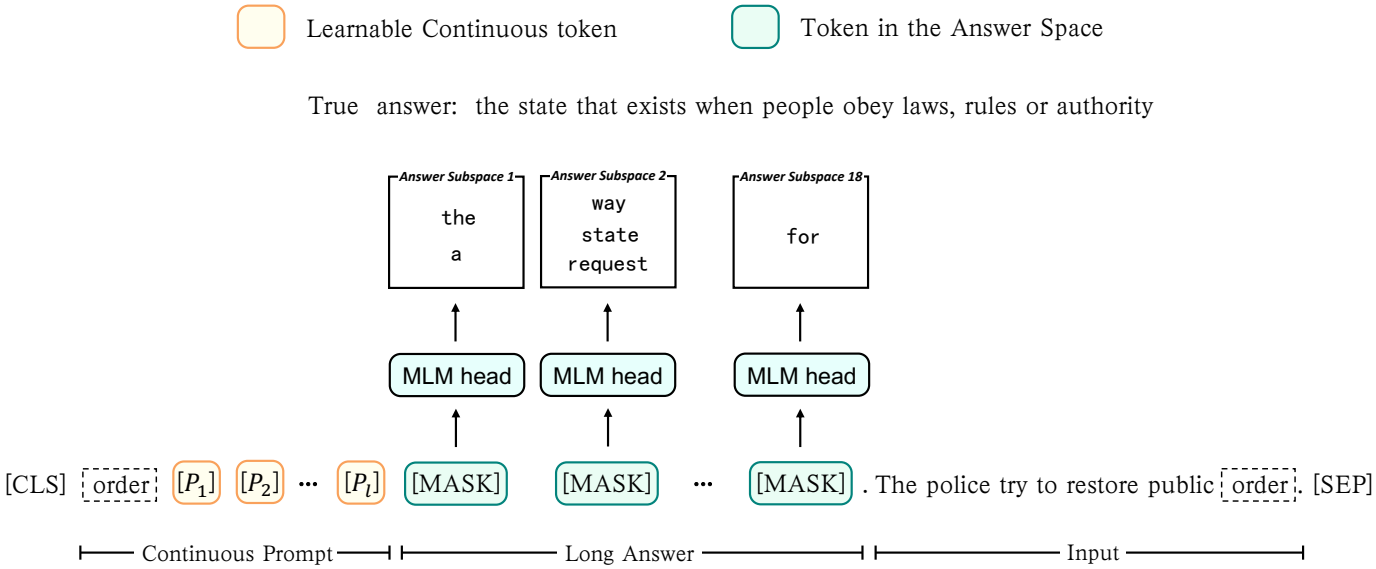

Fig. 1: Illustration of KLAPrompt. There are two steps in KLAPrompt: prompt engineering and answer engineering. In prompt engineering, we use auxiliary virtual tokens $[\mathrm{P}_1], [\mathrm{P}_2], \ldots, [\mathrm{P}_l]$ to replace natural language words. In answer engineering, we split the whole answer space into several answer subspaces according to the token's position in the long answer.

2) We propose a novel long-answer prompt learning method (KLAPrompt), which provide a reasonable solution for two main challenges for answer engineering: (a) When there are many classes, how to seek the proper answer space? (b) How to decode the multi-token answers?
3) Extensive experiments on five Chinese NLP tasks demonstrate the proposed method significantly empowers the widely-adopted pre-trained language models. The empirical studies also confirm that the KLAPrompt with "sense" knowledge gains more significant improvement with less fine-tuning data.
4) We collect a word sense prediction dataset (WSP) based on Xinhua Dictionary, which is available at https://github.com/Xie-Zuotong/WSP

II. Related work

A. Semantic Knowledge

Semantic knowledge contains the meaning of words, phrases, and sentences, examining how meaning is encoded in a language. It has been extensively used in various natural language processing tasks [18]–[21].

ERNIE [22] has improved BERT's masking strategy to integrate entity information in the knowledge graph. In Chinese, an entity or phrase is composed of several Chinese words. If only a single word is masked, the model can easily predict the masked content only through the context information, without paying attention to the composition of phrases and entities, as well as the syntactic and semantic information in sentences. Therefore, ERNIE masks all tokens that compose a whole phrase or entity at the same time. However, the phrase in ERNIE usually consists of two or three tokens. When the number of consecutive tokens exceeds twenty, the model is difficult to train, and the performance will decline. KnowBERT [23] integrates WordNet [24] and a subset of Wikipedia into BERT and uses the Knowledge Attention and Recontextualization mechanism to explicitly model entity spans in the input text. SenseBERT [25] adds a masked-word sense prediction task as an additional task to learn the "sense" knowledge in WordNet. WordNet lexicographers organize all word senses into 45 supersense categories. Hence, it predicts not only the masked words but also their supersenses during pre-training. Both KnowBERT and SenseBERT introduce WordNet into the BERT, but compared with Xinhua Dictionary or Oxford Dictionary, the supersenses in WordNet are relatively limited, and the word meaning is coarse-grained.

Furthermore, most of these methods are training from scratch, which is time-consuming and computationally expensive, making it infeasible for most users.

B. Prompt Learning

Prompt learning is based on the language model used to calculate the probability of text [26]. Unlike adapting pre-trained language models to downstream tasks through objective engineering, prompt learning utilizes additional textual prompts to make downstream tasks look more like those solved during the original language model training.

Radford et al. [27] illustrate that language model can learn NLP tasks without direct supervision, and then prompt learning has gradually become the most popular research direction in natural language processing. Prompt learning includes prompt engineering and answer engineering. For discrete prompts, Brown et al. [28] manually create prefix prompts to deal with diverse natural language processing tasks. For continuous prompts, P-tuning [13] proposes prompts learned by inserting trainable variables into the embedded

input. Recent work [14] manually designed the constrained answer spaces for Named Entity Recognition tasks.

But there are still two challenges for answer engineering: (a) When there are many classes, how to seek the proper answer space? (b) How to decode the multi-token answers?

III. Methodology

In this section, we introduce the KLAPrompt approach and its detailed implementation. There are two steps in our KLAPrompt method: prompt engineering and answer engineering. So we elaborate our method from these two aspects.

A. Prompt Engineering

Prompt engineering, also known as template engineering, is to design a prompting function that results in the most effective performance on the downstream task. It is based on the language model used to calculate the probability of text [29], and it utilizes additional prompts to make downstream tasks look more like those solved during the original language model training.

A template is a textual string with two slots: an input slot [X] for input x and an answer slot [Y]. For example, in the case of sentiment analysis where x = "I love this movie.", the template may take a form such as "[X] Overall, it was a [Y] movie.". Then, the prompt would become "I love this movie. Overall, it was a [Y] movie.". The number of [X] and [Y] slots can be flexibly changed for the need of tasks at hand.

In many cases, these template words are not necessarily composed of natural language tokens; they could be virtual tokens that would be embedded in a continuous space later and optimized through gradient descent.

In our work, we use some auxiliary virtual tokens $[\mathrm{P}_1]$, $[\mathrm{P}_2], \ldots, [\mathrm{P}_l]$, whose parameters are randomly initialized, to make the template more effective, and l is a predefined hyper-parameter. This method performs prompting directly in the embedding space of the model.

The word sense prediction dataset (WSP) contains the word [W], sense y, and sentence x for each example. For the continuous prompt in WSP dataset, we first copy the word [W] mentioned in the sentence x, then add a few auxiliary virtual tokens followed by the answer slot [Y] that the model will predict and the input slot [X]. There is an example of the continuous prompt in Figure 1, and the complete prompt becomes:

$$T(x) = [\mathrm{W}][\mathrm{P}_1], [\mathrm{P}_2], \ldots, [\mathrm{P}_l][\mathrm{Y}].[\mathrm{X}] \tag{1}$$

where $T(\cdot)$ is the template for WSP dataset, [W] is the word mentioned in the input sentence x, $[\mathrm{P}_i]$ is the virtual token, [X] is the input slot for sentence x, and [Y] is the answer slot for sense y. Each embedding of prompts is randomly initialized and optimized during training.

B. Answer Engineering

Unlike prompt engineering, which discovers suitable prompts, answer engineering tries to seek a proper answer space and a map to the original output that brings about an effectual predictive model. For classification-based tasks, there are two main challenges for answer engineering: (a) When there are too many classes, how to select an appropriate answer space becomes a difficult combinatorial optimization problem. (b) When using multi-token answers, how to best decode multiple tokens using PLMs remains unknown [26]. In this section, we propose the long-answer strategy to address the challenges mentioned above.

In prompt learning, for each class $y \in \mathcal{Y}$, the mapping function $\phi(\cdot)$ will map it to the answer $\phi(y) \in \mathcal{V}$, where $\mathcal{V}$ is the answer space. It's easy to find the appropriate answer space and the mapping function when the classes are limited, and all the answer consists of a single token. Unfortunately, there are massive classes in WSP dataset (It includes 7,390 words and 16,495 senses; each word has one to thirteen senses), and the answer is quite long sometimes. Take the word "$order$" as an example. The template and the label word set can be formalized as:

$$\begin{aligned} T(x) &= [\mathrm{W}][\mathrm{P}_1], [\mathrm{P}_2], \ldots, [\mathrm{P}_l][\mathrm{MASK}].\ x \\ \mathcal{V}_{[\mathrm{MASK}]} &= \{\text{“}the\ \ way\ \ in\ \ which\ \ people\ \ or\ \ \ldots\text{”}, \\ &\qquad \text{“}the\ \ state\ \ that\ \ exists\ \ when\ \ \ldots\text{”}, \\ &\qquad \text{“}a\ \ request\ \ for\ food\ \ or\ \ drinks\ \ldots\text{”}\} \end{aligned} \tag{2}$$

But the pre-trained language model like BERT [29] cannot predict the whole long answer at once. So in our work, we split the answer space $\mathcal{V}_{[\mathrm{MASK}]}$ into several answer subspaces $\{\mathcal{V}_{[\mathrm{MASK}]_1}, \mathcal{V}_{[\mathrm{MASK}]_2}, \ldots, \mathcal{V}_{[\mathrm{MASK}]_j}, \ldots, \mathcal{V}_{[\mathrm{MASK}]_n}\}$ according to the token's position in the answer, where n is the length of the answer, and $\phi_j(\mathrm{y})$ is to map the class y to the set of label words $\mathcal{V}_{[\mathrm{MASK}]_j}$ for the j-th masked position $[\mathrm{MASK}]_j$. Here we still take the word "$order$" as an example. As shown in Figure 1, the template and the label word set can be formalized as:

$$\begin{aligned} T(x) &= [\mathrm{W}][\mathrm{P}_1], \ldots, [\mathrm{P}_l][\mathrm{MASK}]_1, \ldots, [\mathrm{MASK}]_n.\ x \\ \mathcal{V}_{[\mathrm{MASK}]_1} &= \{\text{“}the\text{”},\ \ \text{“}a\text{”}\} \\ \mathcal{V}_{[\mathrm{MASK}]_2} &= \{\text{“}way\text{”},\ \ \text{“}state\text{”},\ \ \text{“}request\text{”}\} \\ \mathcal{V}_{[\mathrm{MASK}]_3} &= \{\text{“}in\text{”},\ \ \text{“}that\text{”},\ \ \text{“}for\text{”}\} \\ \mathcal{V}_{[\mathrm{MASK}]_4} &= \{\text{“}which\text{”},\ \ \text{“}exists\text{”},\ \ \text{“}food\text{”}\} \\ &\ldots \end{aligned} \tag{3}$$

In a conventional supervised learning system for natural language processing, we take an input $x \in \mathcal{X}$ and predict an output $y \in \mathcal{Y}$ based on the language model $p(y|x)$. As the template may contain multiple [MASK] tokens, we must consider all masked positions to make predictions, i.e.,

$$p(y|x) = \prod_{j=1}^{n} p([\mathrm{MASK}]_j = \phi_j(\mathrm{y})|T(x)) \tag{4}$$

where n is the number of masked positions in $T(x)$, and $\phi_j(\mathrm{y})$ is to map the class y to the set of label words $\mathcal{V}_{[\mathrm{MASK}]_j}$ for the j-th masked position $[\mathrm{MASK}]_j$. Equation 4 can be used to tune PLMs and classify classes.

Models	STS-B	Book Review	XNLI	Chnsenticorp	IFLYTEK
BERT	50.75	86.62	76.8	93.3	60.52
BERT + KLAPrompt	52.92 ↑	88.63 ↑	78.61 ↑	94.82 ↑	61.58 ↑
RoBERTa	48.23	89.08	78.37	94.85	60.44
RoBERTa + KLAPrompt	50.37 ↑	**91.12** ↑	80.69 ↑	95.1 ↑	61.46 ↑
MacBERT	52.92	88.78	79.05	94.98	60.82
MacBERT + KLAPrompt	**54.67** ↑	90.1 ↑	**81.49** ↑	**95.79** ↑	**62.01** ↑

TABLE II: Experiment results of baselines and our methods on five datasets (Acc.%). "+ KLAPrompt" means that we train PLMs with KLAPrompt method via semantic knowledge infusion training before fine-tuning.

With the pre-trained language model predicting the masked tokens, the loss function of KLAPrompt is given by:

$$\begin{aligned}\mathcal{L} &= -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\log p(y|x) \\ &= -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\log\prod_{j=1}^{n} p([\text{MASK}]_j = \phi_j(\text{y})|T(x))\end{aligned} \tag{5}$$

IV. Experiments

In this section, we present the details of implementation and conduct experiments on five Chinese NLP datasets to evaluate the efficiency and effectiveness of our approach.

A. Datasets

STS-B. This Chinese version of the dataset [1] is translated from the original English dataset STS-B [30] and partially manually revised. Semantic Textual Similarity (STS) measures the meaning similarity of sentences.

Book Review. Book Review dataset [31] is collected from Douban, a Chinese online review website that provides information about books, movies, and music. It's a one-sentence text classification dataset.

XNLI. In our experiment, only the Chinese part of the Cross-language Natural Language Inference dataset (XNLI) [32] is retained. In XNLI, the model should read the two sentences and determine whether the relationship between them is "Entailment", "Contradiction", or "Neutral".

Chnsenticorp. Chnsenticorp [31] is a sentiment analysis dataset which contains 12,000 hotel reviews. 6,000 reviews are positive, and the other 6,000 reviews are negative.

IFLYTEK. The IFLYTEK [33] dataset has more than 17,000 long texts about the application description, including various application topics related to daily life with a total of 119 categories.

Datasets above are with 8.05K, 40.0K, 40.0K, 12.0K, and 17.3K samples respectively. We follow the evaluation metrics and setting used in [31], [33].

[1]https://github.com/pluto-junzeng/CNSD

Models	XNLI
BERT	76.8
- BERT + WSP†	77.34 (+0.54)
- BERT + Continuous Prompt†	77.72 (+0.92)
- BERT + Long-answer Strategy†	78.17 (+1.37)
BERT + KLAPrompt†	**78.61** (+1.81)

TABLE III: Ablation study on XNLI dataset (Acc.%). "+ WSP" means that we train BERT on WSP dataset without the KLAPrompt approach. † means that we train these models on WSP dataset before fine-tuning.

B. Implementation Details

KLAPrompt is based on pre-trained language models. In this work, we choose BERT [29], RoBERTa [34], and MacBERT [35] as our basic models. For all these models, the number of layers is 12, the hidden size is 768, the number of heads is 12, and it contains 110M parameters. These models are optimized by Adam optimizer [36] with the initial learning rate of 1e-5. The training batch size is 64. Each model is trained for 10 epochs and evaluated on the validation set for every epoch. All experiments are carried out using a single NVIDIA GeForce RTX 3090 24GB card.

C. Main Results

The experimental results on the development set of five Chinese natural language processing datasets are presented in Table II. We show each original model and the model trained with KLAPrompt method (e.g., BERT and BERT + KLAPrompt). We find that all pre-trained language models trained with KLAPrompt method have achieved significant improvement compared to the original PLMs. For STS-B, Book Review, and XNLI datasets, RoBERTa + KLAPrompt pushes up the final results by 2.14%, 2.04%, and 2.32%. And for IFLYTEK dataset, the method still can raise the accuracy by more than 1%. This superior performance proves that infusing

Setting	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$
BERT + KLAPrompt	94.24	**94.82**	94.65	94.57	94.33

TABLE IV: Model performance on Chnsenticorp dataset (Acc.%) w.r.t. different values of hyper-parameter l.

external semantic knowledge by KLAPrompt approach can empower the widely-adopted pre-trained language models.

D. Ablation Study

In our proposed KLAPrompt, two components may affect the performance: Continuous Prompt and Long-answer Strategy. To explore such effects, we conduct an ablation experiment using the XNLI dataset. We first compare BERT with BERT + WSP to showcase the advantages of external semantic knowledge in WSP dataset. BERT + WSP is trained on WSP dataset with its original masked language model (MLM), and it does not use the KLAPrompt method. Experimental results demonstrate that introducing semantic information in Xinhua Dictionary can consistently improve language modeling and downstream tasks. Then we explore the effects of Continuous Prompt and Long-answer Strategy. As shown in Table III, both Continuous Prompt and Long-answer Strategy can improve performance on this Natural Language Inference dataset. In addition, the improvement brought by using Continuous Prompt or Long-answer Strategy alone is less than using the whole KLAPrompt method.

The hyper-parameter l is the number of virtual tokens in the continuous prompt. To explore its impact on the performance of KLAPrompt, we test with different values of hyper-parameter $l = \{1, 2, 3, 4, 5\}$. As shown in Table IV, we can see that the performance of the model shows a trend of rising at first and then falling as l increases. Especially when $l = 2$, the model has the best performance.

We also investigate the consistent improvements with different percentages of downstream training data. The experiment results in Figure 2 illustrate that the improvement is more obvious when the amount of data is smaller. In other words, KLAPrompt with semantics knowledge can benefit data-scarce downstream tasks. Because when the training data is limited, the task depends on the pre-trained language model and the additional semantics knowledge.

V. Conclusion

In this work, we propose the KLAPrompt approach to introduce semantics knowledge into pre-trained language models. What's more, we collect a word sense prediction dataset (WSP). Extensive experiments on five Chinese NLP datasets show the effectiveness of KLAPrompt method in integrating semantic knowledge. For future work, we will infuse commonsense information, domain-specific information, and knowledge graphs into the pre-trained language models.

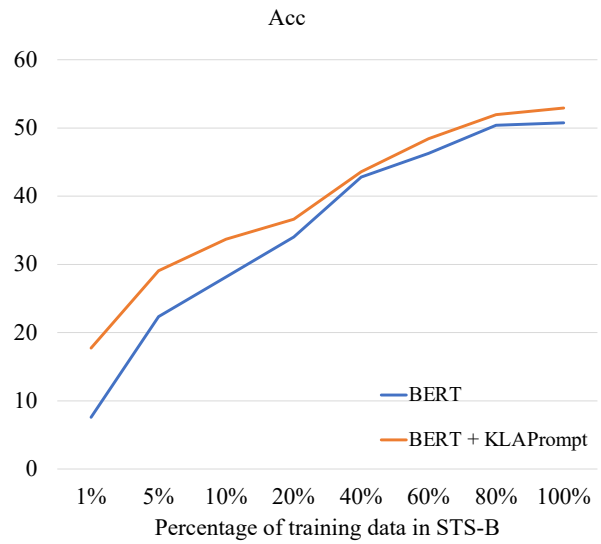

Fig. 2: Performance of BERT and BERT + KLAPrompt method with different amounts of training data.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (Grant No.62276154), Research Center for Computer Network (Shenzhen) Ministry of Education, Beijing Academy of Artificial Intelligence (BAAI), the Natural Science Foundation of Guangdong Province (Grant No. 2023A1515012914), Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033 and JSGG20210802154402007), and the Major Key Project of PCL for Experiments and Applications (PCL2021A06).

References

[1] Xiaocheng Feng, Xiachong Feng, Bing Qin, Zhangyin Feng, and Ting Liu, "Improving low resource named entity recognition using cross-lingual knowledge transfer.," in *IJCAI*, 2018, vol. 1, pp. 4071–4077.

[2] Songming Zhang, Ying Zhang, Yufeng Chen, Du Wu, Jinan Xu, and Jian Liu, "Exploiting morpheme and cross-lingual knowledge to enhance mongolian named entity recognition," *Transactions on Asian and Low-Resource Language Information Processing*, 2022.

[3] Pan Liu, Yanming Guo, Fenglei Wang, and Guohui Li, "Chinese named entity recognition: The state of the art," *Neurocomputing*, vol. 473, pp. 37–53, 2022.

[4] Xing Liu, Huiqin Chen, and Wangui Xia, "Overview of named entity recognition," *Journal of Contemporary Educational Research*, vol. 6, no. 5, pp. 65–68, 2022.

[5] Ziran Li, Ning Ding, Zhiyuan Liu, Haitao Zheng, and Ying Shen, "Chinese relation extraction with multi-grained information and external linguistic knowledge," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4377–4386.

[6] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig, "Probing linguistic features of sentence-level representations in neural relation extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1534–1545.

[7] Bo Qiao, Zhuoyang Zou, Yu Huang, Kui Fang, Xinghui Zhu, and Yiming Chen, "A joint model for entity and relation extraction based on bert," *Neural Computing and Applications*, pp. 1–11, 2022.

[8] Huiyu Sun and Ralph Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 861–870, 2022.

[9] Zhiqiang Yu, Yantuan Xian, Zhengtao Yu, Yuxin Huang, and Junjun Guo, "Linguistic feature template integration for chinese-vietnamese neural machine translation," *Frontiers of Computer Science*, vol. 16, no. 3, pp. 1–3, 2022.

[10] Zhiqiang Yu, Zhengtao Yu, Yantuan Xian, Yuxin Huang, and Junjun Guo, "Improving chinese-vietnamese neural machine translation with linguistic differences," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 2, pp. 1–12, 2022.

[11] Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch, "Survey of low-resource machine translation," *Computational Linguistics*, vol. 48, no. 3, pp. 673–732, 2022.

[12] Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur, "Neural machine translation for low-resource languages: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, 2023.

[13] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang, "Gpt understands, too," *arXiv preprint arXiv:2103.10385*, 2021.

[14] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang, "Template-based named entity recognition using bart," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 1835–1845.

[15] Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Marzieh Saeidi, Lambert Mathias, Veselin Stoyanov, and Majid Yazdani, "Perfect: Prompt-free and efficient few-shot learning with language models," in *Proceedings Of The 60Th Annual Meeting Of The Association For Computational Linguistics (Acl 2022), Vol 1:(Long Papers)*. ASSOC COMPUTATIONAL LINGUISTICS-ACL, 2022, number CONF, pp. 3638–3652.

[16] Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu, "Prototypical verbalizer for prompt-based few-shot tuning," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 7014–7024.

[17] Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, Roberto Navigli, et al., "Recent trends in word sense disambiguation: A survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conference on Artificial Intelligence, Inc, 2021.

[18] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum, "Linguistically-informed self-attention for semantic role labeling," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 5027–5038.

[19] Dehui Kong, Xiliang Li, Shaofan Wang, Jinghua Li, and Baocai Yin, "Learning visual-and-semantic knowledge embedding for zero-shot image classification," *Applied Intelligence*, pp. 1–15, 2022.

[20] Sebastian Kiefer, "Case: Explaining text classifications by fusion of local surrogate explanation models with contextual and semantic knowledge," *Information Fusion*, vol. 77, pp. 184–195, 2022.

[21] Gabriel Grand, Idan Asher Blank, Francisco Pereira, and Evelina Fedorenko, "Semantic projection recovers rich human knowledge of multiple object features from word embeddings," *Nature human behaviour*, vol. 6, no. 7, pp. 975–987, 2022.

[22] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu, "Ernie: Enhanced representation through knowledge integration," *arXiv preprint arXiv:1904.09223*, 2019.

[23] Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith, "Knowledge enhanced contextual word representations," 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)*, 2019, pp. 43–54.

[24] George A Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[25] Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham, "Sensebert: Driving some sense into bert," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4656–4667.

[26] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *arXiv preprint arXiv:2107.13586*, 2021.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9, 2019.

[28] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "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)*, 2019, pp. 4171–4186.

[30] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia, "Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation," *arXiv preprint arXiv:1708.00055*, 2017.

[31] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang, "K-bert: Enabling language representation with knowledge graph," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 2901–2908.

[32] Alexis Conneau, Ruty Rinott, Guillaume Lample, Holger Schwenk, Ves Stoyanov, Adina Williams, and Samuel R Bowman, "Xnli: Evaluating cross-lingual sentence representations," in *2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*. Association for Computational Linguistics, 2020, pp. 2475–2485.

[33] Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al., "Clue: A chinese language understanding evaluation benchmark," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 4762–4772.

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[35] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu, "Revisiting pre-trained models for chinese natural language processing," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 657–668.

[36] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd international conference for learning representations, San Diego*, 2015.