electronics

MDPI

Article

Prompt Learning with Structured Semantic Knowledge Makes Pre-Trained Language Models Better

Hai-Tao Zheng [1,2,*], Zuotong Xie [1], Wenqiang Liu [3], Dongxiao Huang [3], Bei Wu [3] and Hong-Gee Kim [4]

[1] Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China; xiezt20@mails.tsinghua.edu.cn
[2] Pengcheng Laboratory, Shenzhen 518055, China
[3] Interactive Entertainment Group, Tencent Inc., Shenzhen 518057, China; masonqliu@tencent.com (W.L.); donxhuang@tencent.com (D.H.); bellabwu@tencent.com (B.W.)
[4] School of Dentistry, Seoul National University, Seoul 03080, Republic of Korea; hgkim@snu.ac.kr
* Correspondence: zheng.haitao@sz.tsinghua.edu.cn

Abstract: Pre-trained language models with structured semantic knowledge have demonstrated remarkable performance in a variety of downstream natural language processing tasks. The typical methods of integrating knowledge are designing different pre-training tasks and training from scratch, which requires high-end hardware, massive storage resources, and long computing times. Prompt learning is an effective approach to tuning language models 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), with three different long-answer strategies, to incorporate semantic knowledge into pre-trained language models, and we compare the performance of these three strategies through experiments. We also explore the effectiveness of the KLAPrompt method in the medical field. Additionally, we generate a word sense prediction dataset (WSP) based on the Xinhua Dictionary and a disease and category prediction dataset (DCP) based on MedicalKG. Experimental results show that discrete answers with the answer space partitioning strategy achieve the best results, and introducing structured semantic information can consistently improve language modeling and downstream tasks.

Keywords: prompt learning; semantic knowledge; pre-trained language model

check for updates

Citation: Zheng, H.-T.; Xie, Z.; Liu, W.; Huang, D.; Wu, B.; Kim, H.-G. Prompt Learning with Structured Semantic Knowledge Makes Pre-Trained Language Models Better. *Electronics* **2023**, *12*, 3281. https://doi.org/10.3390/electronics12153281

Academic Editor: Heung-Il Suk

Received: 26 June 2023
Revised: 13 July 2023
Accepted: 20 July 2023
Published: 30 July 2023

1. Introduction

In recent years, pre-trained language models (PLMs) such as BERT [1], XLNet [2], and RoBERTa [3] have achieved promising results in many natural language processing (NLP) tasks [4–6]. Although no explicit syntactic rules and concepts are introduced, these models can perform well with extensive pre-training on large-scale unlabeled corpora in various self-supervised ways. Nevertheless, recent works illustrate that external semantic knowledge can improve downstream NLP tasks, including named entity recognition [7,8], relation extraction [9,10], and machine translation [11–13]. However, traditional approaches to introducing knowledge involve 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 [14–17], and it can also be used to integrate knowledge.

The Xinhua Dictionary, the most authoritative and influential modern Chinese dictionary, contains massive and comprehensive content, such as word forms, pronunciation, precise definitions, and rich examples. In contrast to WordNet [18] and HowNet [19], the “senses” in the Xinhua Dictionary are more abundant, detailed, and fine-grained. It can offer strong and efficient support for language models to understand Chinese word semantics. In the Xinhua Dictionary, the “sense” is the meaning of the word [20], and a Chinese word can have more than one sense. An example from the Xinhua Dictionary is shown in Table 1. The "sense" is composed of a long string of tokens, but the typical methods of prompt learning accept one token as the answer. Thus, the issue of how to properly use the wealth of long-answer information is a challenging problem.

Table 1. An example of a word and its senses and phrases in the Xinhua Dictionary.

Word	Sense	Phrase
order	ID:06029 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
	ID:06030 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
	ID:06031 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

To address this challenge, we propose the long-answer prompt learning method (KLAPrompt), with three different long-answer strategies, and collect a word sense prediction dataset (WSP) based on the Xinhua Dictionary to introduce fine-grained semantic knowledge. According to the different forms of answers, they can be divided into three strategies: discrete answers, continuous answers, and sentence similarity. In the discrete answer strategy, 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 1 are $\{``the", ``a"\}$, $\{``way", ``state", ``request"\}$, $\{``in", ``that", ``for"\}$, $\{``which", ``exists", ``food"\}, \ldots, \{``for"\}$. Then, we train pre-trained language models on the WSP dataset to predict the sense, and each word of the sense will be predicted independently. In the continuous answer strategy, we use virtual answer tokens, which can be optimized through gradient descent, to replace the natural language in discrete answers. Due to the many senses in the Xinhua Dictionary, we assign several virtual tokens to each sense and optimize the token embeddings for each sense together with prompt token embeddings. In the sentence similarity strategy, we average the embeddings of the masked part in the masked language model's (MLM) output, calculate the cosine similarity between this and the sentence embedding of the original long answer, and then maximize the similarity during the training procedure.

Furthermore, we explore the effectiveness of the KLAPrompt method in the medical field. Firstly, we collect a disease and category prediction dataset (DCP) based on MedicalKG (https://github.com/zhihao-chen/QASystemOnMedicalKG accessed on 28 April 2023), which contains specific disease knowledge such as descriptions, departments, symptoms, causes, prevention, checking items, recommended foods, and recommended drugs. Then, we apply the KLAPrompt method to introduce fine-grained disease knowledge into the pre-trained language models.

We conduct comprehensive experiments on five open-domain NLP datasets and five health-related datasets. Experimental results demonstrate that pre-trained language models achieve superior performance based on the strength of the semantic knowledge in the Xinhua Dictionary and the disease knowledge in MedicalKG. Empirical studies also verify that KLAPrompt with the discrete answer strategy is the best method to integrate structured semantic knowledge into pre-trained language models.

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

(1) We introduce more abundant and fine-grained semantic knowledge from the Xinhua Dictionary and disease knowledge from MedicalKG into pre-trained language models, enhancing the models' ability to understand Chinese word semantics and medical science.

(2) We propose a novel long-answer prompt learning method (KLAPrompt), which provides a reasonable solution to two main challenges in answer engineering: (a) When there are many classes, how can we seek the proper answer space? (b) How can we decode the multi-token answers?

(3) Extensive experiments on five Chinese NLP datasets and five biomedical datasets demonstrate that the proposed method significantly empowers the widely adopted pre-trained language models. The empirical studies also confirm that KLAPrompt with the discrete answer strategy is the best method to integrate structured semantic knowledge.

(4) We generate a word sense prediction dataset (WSP) based on the Xinhua Dictionary, which is available at https://github.com/Xie-Zuotong/WSP (accessed on 28 April 2023). We also collect a disease and category prediction dataset (DCP) based on MedicalKG, which is available at https://github.com/Xie-Zuotong/DCP (accessed on 28 April 2023).

The rest of the paper is organized as follows. Section 2 briefly reviews the existing methods of integrating semantic knowledge, previous approaches in prompt learning, and several biomedical pre-trained language models. In Section 3, we introduce the KLAPrompt approach and its application in the medical field. Section 4 shows the experimental results on five Chinese NLP datasets. Section 5 presents experimental studies on five biomedical datasets. Finally, the conclusions of this research are drawn in Section 6.

2. Related Works

2.1. Semantic Knowledge

Semantic knowledge includes the meanings of words, phrases, and sentences, examining how meaning is encoded in a language. It has been extensively used in various natural language processing tasks [21–24]. ERNIE [25] has improved BERT's masking strategy to integrate entity information into 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, a 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 [26] integrates WordNet [18] 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 [27] 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 BERT, but, compared with the Xinhua Dictionary or Oxford Dictionary, the supersenses in WordNet are relatively limited, and the word meaning is coarse-grained. Furthermore, most of these methods require training from scratch, which is time-consuming and computationally expensive, making it infeasible for most users.

2.2. Prompt Learning

Prompt learning is based on the language model used to calculate the probability of text [28]. Unlike adapting pre-trained language models to downstream tasks through objective engineering, prompt learning utilizes additional textual prompts to make downstream tasks resemble those solved during the original language model training. Radford et al. [29] illustrate that language models can learn NLP tasks without direct supervision, and 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. [30] manually created prefix prompts to deal with diverse natural language processing tasks. For continuous prompts, P-tuning [14] proposes prompts learned by inserting trainable variables into the embedded input. A recent work [15] manually designed the constrained answer spaces for named entity recognition tasks. However, there are still two challenges in answer engineering: (a) When there are many classes, how can we seek the proper answer space? (b) How can we decode the multi-token answers?

2.3. Biomedical PLMs

Biomedical PLMs are typically built by adapting a general-domain PLM to the biomedical domain with the same model architecture and training objectives, as exemplified by BioBERT [31], BlueBERT [32], SciBERT [33], ClinicalBERT [34], and PubMedBERT [35]. BioBERT is the first model to utilize continuous pre-training on biomedical domain corpora. BlueBERT was pre-trained on PubMed abstracts and MIMIC-III clinical notes, and then evaluated on the Biomedical Language Understanding Evaluation (BLUE) benchmark. ClinicalBERT uses clinical notes, including lab values and medications, instead of plaintext data based on BERT. Moreover, PubMedBERT learns model weights from scratch via a large-scale training corpus.

While great efforts have been made to build English biomedical PLMs, there are only a few studies discussing building biomedical PLMs in Chinese, such as MC-BERT [36], MedBERT (https://github.com/trueto/medbert accessed on 15 April 2023) and SMedBERT [37], derived from a general-domain BERT, with the latter two further developed in a knowledge-enhanced manner. MC-BERT proposes entity masking and phrase masking strategies in a coarse-grained context to learn the medical word representations from a medical corpus, while neglecting the internal relations of medical entities. The MedBERT model was pre-trained on 650 million Chinese clinical natural language texts. SMedBERT proposes the mention-neighbour hybrid attention to learn heterogeneous entity information, which infuses the semantic representations of entity types into the homogeneous neighboring entity structure.

3. Methodology

In this section, we introduce the KLAPrompt approach and its application in the medical field. There are two steps in our KLAPrompt method: prompt engineering and answer engineering. Thus, we elaborate on our method from these two aspects.

3.1. Prompt Engineering

Prompt engineering, also known as template engineering, aims to design a prompting function that results in the most effective performance in a downstream task. There are two main categories of prompts: discrete prompts and continuous prompts. In this section, we construct these prompts for the word sense prediction dataset (WSP) to introduce the semantic knowledge from the Xinhua Dictionary.

3.1.1. Discrete Prompts

Discrete prompts are composed of the alignment of words in natural language. The most common method to create discrete prompts is to manually create crafted templates to handle different tasks. 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 discrete prompt would become "I love this movie. Overall, it was a [Y] movie".

The word sense prediction dataset (WSP) contains one word, phrase, sense, and sentence for each example. For discrete prompts, we first copy the word [C] mentioned in the sentence [X], and then add a few natural language words followed by the sense [Y] that the model will predict. The number of [X] slots and the number of [Y] slots can be flexibly changed according to the needs of the task at hand. The detailed composition of [Y] will be described in Section 3.2. The manual templates are as follows:

$$\begin{aligned} T_1(x) &= \text{The meaning of [C] is [Y]. } [X] \\ T_2(x) &= \text{The meaning of “[C]” is [Y]. } [X] \\ T_3(x) &= \text{Among them, the meaning of [C] is [Y]. } [X] \\ T_4(x) &= \text{In this sentence, the meaning of [C] is [Y]. } [X] \end{aligned}$$

An example of a discrete prompt is given in Figure 1, where [C] is “order”, the form of [Y] is “The state that exists when people obey laws, rules or authority”, and [X] is “The police try to restore public order”.

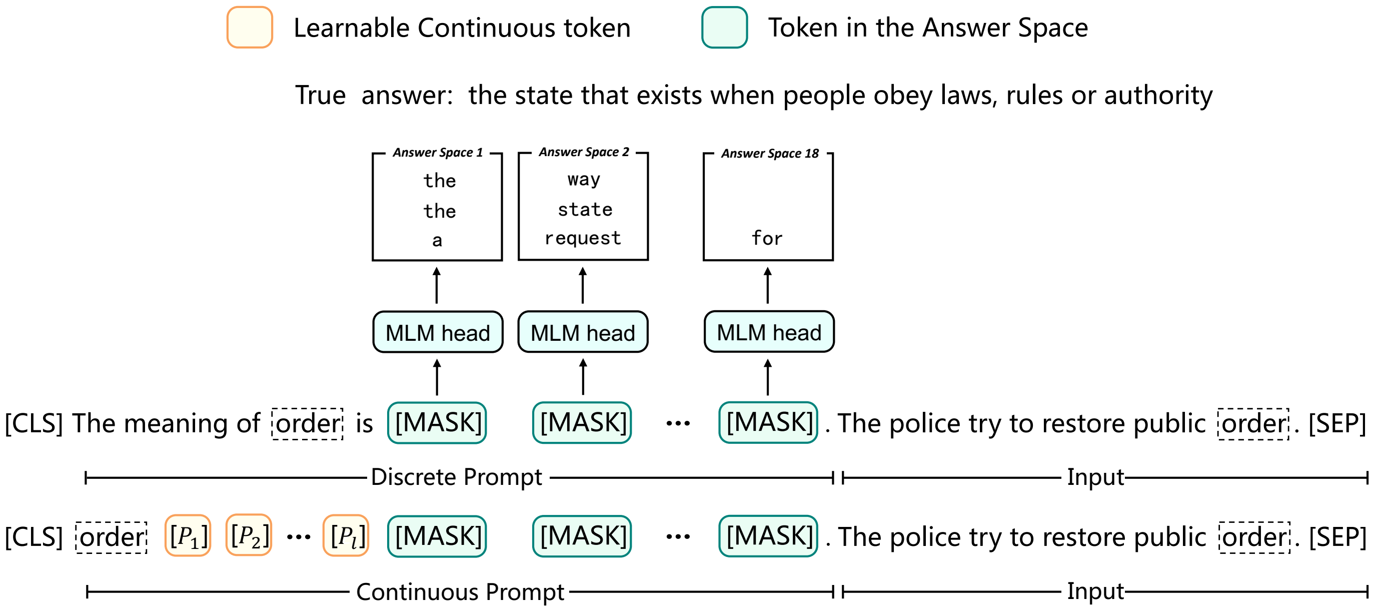

Learnable Continuous token
Token in the Answer Space
True answer: the state that exists when people obey laws, rules or authority
Answer Space 1
the
the
a
Answer Space 2
way
state
request
Answer Space 18
for
MLM head
MLM head
MLM head
[CLS] The meaning of order is [MASK] [MASK] ··· [MASK] . The police try to restore public order. [SEP]
Discrete Prompt
Input
[CLS] order [P_1] [P_2] ··· [P_l] [MASK] [MASK] ··· [MASK] . The police try to restore public order. [SEP]
Continuous Prompt
Input

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

3.1.2. Continuous Prompts

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

For continuous prompts in WSP, we use some auxiliary virtual tokens $[\mathrm{P}_1], [\mathrm{P}_2], \ldots, [\mathrm{P}_l]$ to replace natural language words, where $[\mathrm{P}_i]$ can be [unused1]∼[unused99] in the vocabulary of the pre-trained language model, and l is a predefined hyper-parameter. This method performs prompting directly in the embedding space of the model. An example of a continuous prompt is given in Figure 1, for which the complete prompt becomes

$$T_5(x) = \text{[C] } [\mathrm{P}_1], [\mathrm{P}_2], \ldots, [\mathrm{P}_l] \text{ [Y]. } [X]$$

where $T(\cdot)$ is the template for the WSP dataset, [C] 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.

3.2. 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. There are two main challenges in answer engineering: (a) when there are too many classes, the selection of an appropriate answer space becomes a difficult combinatorial optimization problem; (b) when using multi-token answers, the issue of how to best decode multiple tokens using PLMs remains unresolved [28]. In this section, we propose three independent and different long-answer prompt learning strategies for the word sense prediction dataset (WSP) to integrate the "sense" knowledge from the Xinhua Dictionary.

3.2.1. Discrete Answers

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 is easy to find the appropriate answer space and the mapping function when the classes are limited, and all the answers consist of a single token. Unfortunately, there is a massive number of classes in the WSP dataset (it includes 7390 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 follows:

$$\begin{aligned} T(x) &= [\mathrm{C}][\mathrm{P}_1], [\mathrm{P}_2], \ldots, [\mathrm{P}_l][\mathrm{MASK}].\ x \\ \mathcal{V}_{[\mathrm{MASK}]} &= \{\text{"}the\ \ way\ \ in\ \ which\ \ people\ \ or\ \ \ldots\text{"}, \\ &\quad\ \text{"}the\ \ state\ \ that\ \ exists\ \ when\ \ \ldots\text{"}, \\ &\quad\ \text{"}a\ \ request\ \ for\ food\ \ or\ \ drinks\ \ldots\text{"}\} \end{aligned} \tag{1}$$

However, a pre-trained language model such as BERT [1] cannot predict the whole long answer at once. Thus, 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 follows:

$$\begin{aligned} T(x) &= [\mathrm{C}][\mathrm{P}_1], \ldots, [\mathrm{P}_l][\mathrm{MASK}]_1, \ldots, [\mathrm{MASK}]_n.\ x \\ \mathcal{V}_{[\mathrm{MASK}]_1} &= \{\text{"}the\text{"},\ \text{"}a\text{"}\} \\ \mathcal{V}_{[\mathrm{MASK}]_2} &= \{\text{"}way\text{"},\ \text{"}state\text{"},\ \text{"}request\text{"}\} \\ \mathcal{V}_{[\mathrm{MASK}]_3} &= \{\text{"}in\text{"},\ \text{"}that\text{"},\ \text{"}for\text{"}\} \\ \mathcal{V}_{[\mathrm{MASK}]_4} &= \{\text{"}which\text{"},\ \text{"}exists\text{"},\ \text{"}food\text{"}\} \\ &\ldots \end{aligned} \tag{2}$$

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.,

$$p(y|x) = \prod_{j=1}^{n} p([\mathrm{MASK}]_j = \phi_j(\mathrm{y})|T(x)) \tag{3}$$

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 (3) can be used to tune PLMs and classify classes.

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

$$\begin{aligned}\mathcal{L} &= -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\log p(y|x) \\ &= -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\log\prod_{j=1}^{n} p([\mathrm{MASK}]_j = \phi_j(\mathrm{y})|T(x))\end{aligned} \tag{4}$$

3.2.2. Continuous Answers

In contrast to discrete answers, continuous answers use virtual answer tokens optimized directly in the embedding space. WARP [38] utilizes a virtual token for each class label and optimizes the token embedding for each class together with prompt token embeddings.

However, for the word sense prediction dataset (WSP), we do not have 16,495 unused virtual tokens in the vocabulary for 16,495 classes. Thus, we design the answer space according to the sense ID in the Xinhua Dictionary. The continuous answers consist of five virtual tokens, and each token belongs to the answer space $Q = \{[\mathrm{Q}_0], [\mathrm{Q}_1], \ldots, [\mathrm{Q}_9]\}$, where $[\mathrm{Q}_i]$ can be [unused1]∼[unused99] in the vocabulary of the pre-trained language model, which would be embedded in a continuous space later and optimized through gradient descent. In Figure 2, the sense ID is "06030", and the true answer is "$[\mathrm{Q}_0][\mathrm{Q}_6][\mathrm{Q}_0][\mathrm{Q}_3][\mathrm{Q}_0]$". In this way, each class has a different set of virtual tokens. We can also use more virtual tokens and adopt different strategies.

In our continuous answer method, the "sense" answer is $w = \{w_1, w_2, w_3, w_4, w_5\}$, where $w_j = \phi_j(\mathrm{y}) \in Q \in \mathcal{V}$, 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 $[\mathrm{MASK}]_j$. As we attempt to obtain the predictions of the masked tokens, the objective is similar to Equation (4):

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

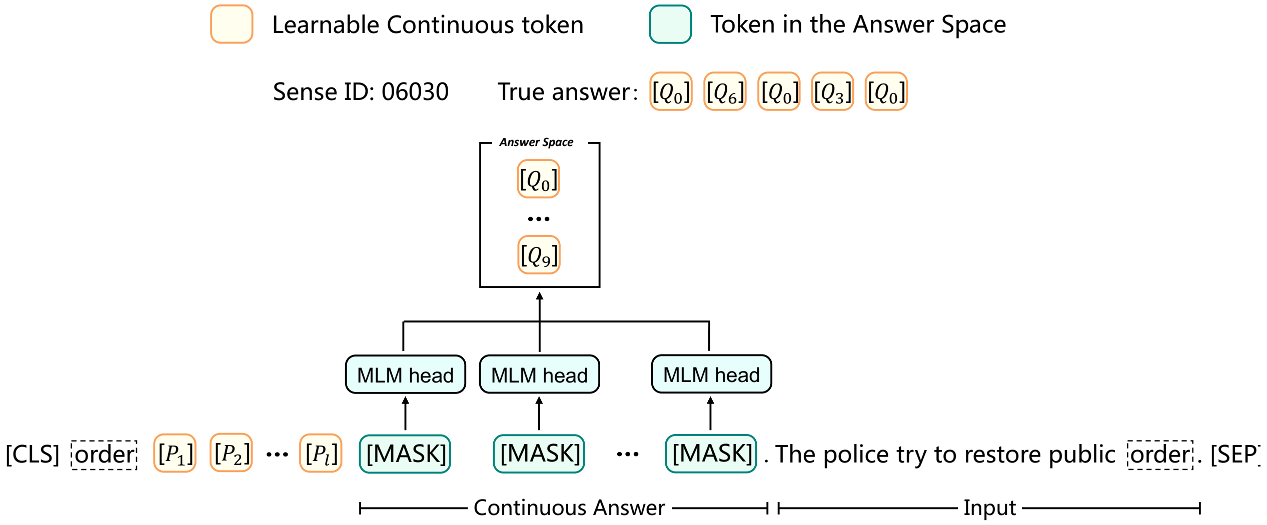

Learnable Continuous token
Token in the Answer Space
Sense ID: 06030
True answer: [Q0] [Q6] [Q0] [Q3] [Q0]
Answer Space
[Q0]
···
[Q9]
MLM head
MLM head
MLM head
[CLS] order [P1] [P2] ··· [Pl] [MASK] [MASK] ··· [MASK] . The police try to restore public order . [SEP
Continuous Answer
Input

Figure 2. Illustration of KLAPrompt with continuous answer strategy. In the continuous answer strategy, we use virtual answer tokens, which can be optimized through gradient descent, to replace the natural language in discrete answers.

3.2.3. Sentence Similarity

Different from the above two methods, we propose another approach to dealing with the problem whereby the masked language model (MLM) is unable to predict the whole long answer at once.

We average the embeddings of the masked part and the original true answer to obtain their sentence embedding. Thus, in this method, the answer space is a sentence embedding space. Then, we maximize the cosine similarity between these two sentence embeddings to make the predicted tokens as similar to the original answer tokens as possible.

When calculating the similarity, the predicted answer and the true answer share the same MLM head to ensure that the two sentence embeddings are generated by the same model. An example of this method is shown in Figure 3.

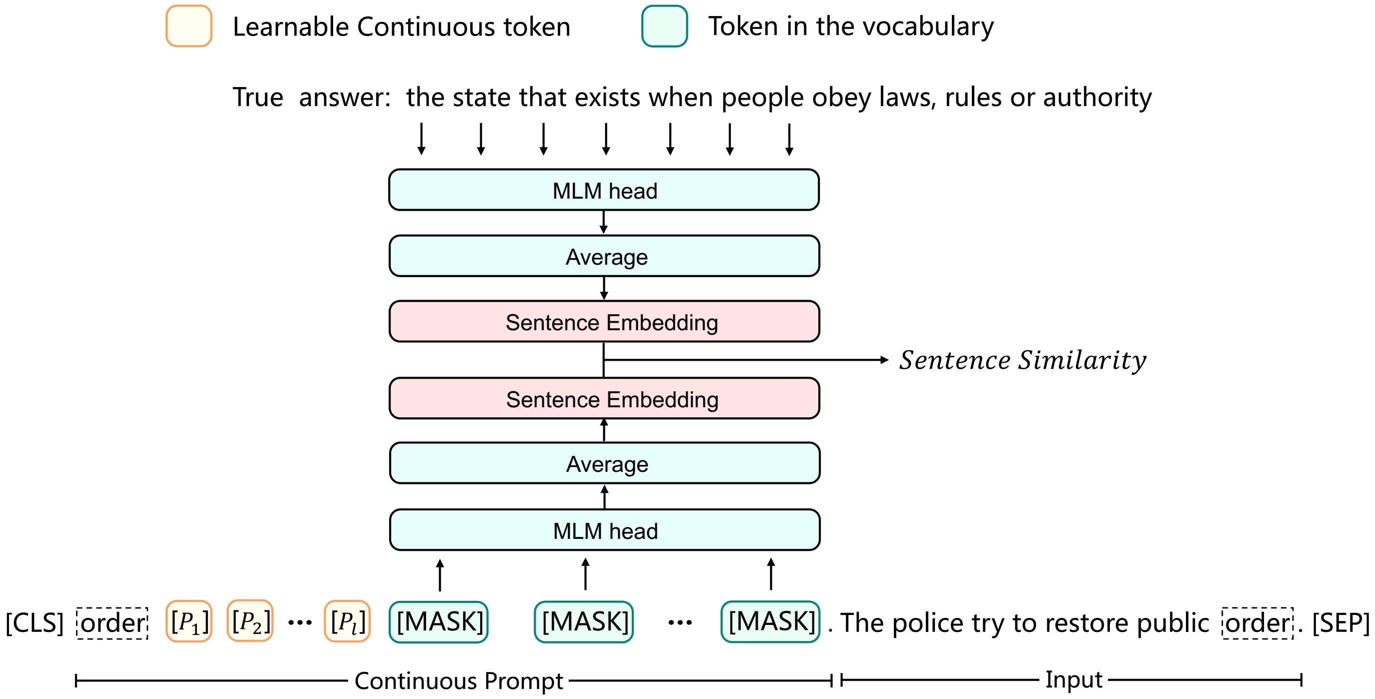
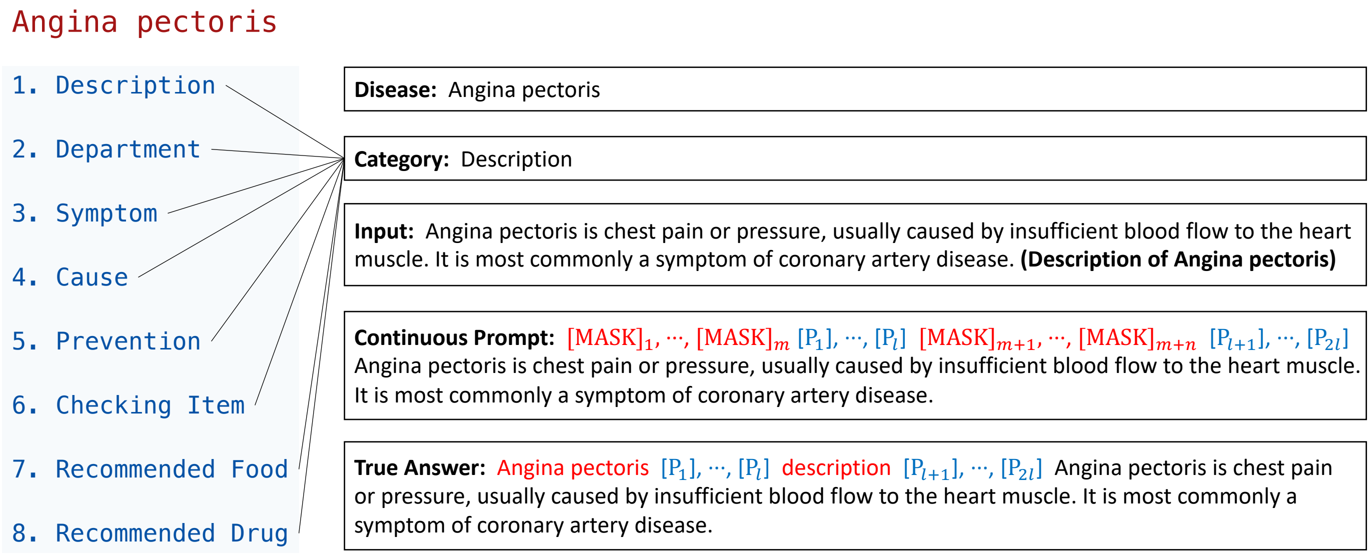

Learnable Continuous token
Token in the vocabulary
True answer: the state that exists when people obey laws, rules or authority
MLM head
Average
Sentence Embedding
Sentence Similarity
Sentence Embedding
Average
MLM head
[CLS] order [P_1] [P_2] ··· [P_l] [MASK] [MASK] ··· [MASK] . The police try to restore public order . [SEP]
Continuous Prompt
Input

Figure 3. Illustration of KLAPrompt with sentence similarity strategy. In the sentence similarity strategy, we average the embeddings of the masked part and the original true answer to obtain their sentence embedding. Then, we maximize the cosine similarity between these two sentence embeddings.

3.3. KLAPrompt's Application in the Medical Field

We also explore the effectiveness of the KLAPrompt method in the medical field. Firstly, we collect a disease and category prediction dataset (DCP) based on MedicalKG (https://github.com/zhihao-chen/QASystemOnMedicalKG accessed on 3 April 2023), which contains specific disease knowledge such as descriptions, departments, symptoms, causes, prevention, checking items, recommended foods, and recommended drugs. Then, we design the continuous prompt for the DCP dataset. An example of a continuous prompt and disease knowledge in MedicalKG is shown in Figure 4.



Disease: Angina pectoris

Category: Description

Input: Angina pectoris is chest pain or pressure, usually caused by insufficient blood flow to the heart muscle. It is most commonly a symptom of coronary artery disease. **(Description of Angina pectoris)**

Continuous Prompt: $[\text{MASK}]_1, \cdots, [\text{MASK}]_m$ $[\text{P}_1], \cdots, [\text{P}_l]$ $[\text{MASK}]_{m+1}, \cdots, [\text{MASK}]_{m+n}$ $[\text{P}_{l+1}], \cdots, [\text{P}_{2l}]$ Angina pectoris is chest pain or pressure, usually caused by insufficient blood flow to the heart muscle. It is most commonly a symptom of coronary artery disease.

True Answer: Angina pectoris $[\text{P}_1], \cdots, [\text{P}_l]$ description $[\text{P}_{l+1}], \cdots, [\text{P}_{2l}]$ Angina pectoris is chest pain or pressure, usually caused by insufficient blood flow to the heart muscle. It is most commonly a symptom of coronary artery disease.

Figure 4. An example of a continuous prompt and disease knowledge in MedicalKG. For each disease, there are eight categories of disease knowledge. We utilize the masked language model (MLM) to predict the disease and category simultaneously.

The continuous prompt for the DCP dataset can be formalized as follows:

$$\begin{aligned} T(x) &= [\text{Y}_1]\ [\text{P}_1], \ldots, [\text{P}_l]\ [\text{Y}_2]\ [\text{P}_{l+1}], \ldots, [\text{P}_{2l}]\ [\text{X}] \\ &= [\text{MASK}]_1, \ldots, [\text{MASK}]_m\ [\text{P}_1], \ldots, [\text{P}_l] \\ &\quad [\text{MASK}]_{m+1}, \ldots, [\text{MASK}]_{m+n}\ [\text{P}_{l+1}], \ldots, [\text{P}_{2l}]\ x \end{aligned} \tag{6}$$

where $T(\cdot)$ is the template for the DCP dataset, $[\text{P}_i]$ is the virtual token, and [X] is the input slot for input x. $[\text{Y}_1]$ is the answer slot for $[\text{MASK}]_1, \ldots, [\text{MASK}]_m$, where m is the length of the disease. $[\text{Y}_2]$ is the answer slot for $[\text{MASK}]_{m+1}, \ldots, [\text{MASK}]_{m+n}$, where n is the length of the category. Each embedding of prompts is randomly initialized and optimized during training.

Finally, we apply the answer space partitioning strategy to the masked language model (MLM) to predict the disease and category simultaneously.

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. Take the disease *"Angina pectoris"* and category *"description"* as an example. The template and the label word set can be formalized as follows:

$$\begin{aligned} T(x) &= [\text{Y}_1]\ [\text{P}_1], \ldots, [\text{P}_l]\ [\text{Y}_2]\ [\text{P}_{l+1}], \ldots, [\text{P}_{2l}]\ [\text{X}] \\ \mathcal{V}_{[\text{Y}_1]} &= \{\text{"}Diabetes\text{"}, \\ &\qquad \text{"}Angina\ \ pectoris\text{"}, \\ &\qquad \text{"}Iron\ \ deficiency\ \ anemia\text{"} \\ &\qquad \text{"}\ldots\text{"}\} \\ \mathcal{V}_{[\text{Y}_2]} &= \{\text{"}cause\text{"}, \\ &\qquad \text{"}description\text{"}, \\ &\qquad \text{"}checking\ \ item\text{"} \\ &\qquad \text{"}\ldots\text{"}\} \end{aligned} \tag{7}$$

For the DCP dataset, we split the answer space $\mathcal{V}_{[\text{Y}_1]}$ into several answer subspaces $\{\mathcal{V}_{[\text{MASK}]_1}, \mathcal{V}_{[\text{MASK}]_2}, \ldots, \mathcal{V}_{[\text{MASK}]_m}\}$ and the answer space $\mathcal{V}_{[\text{Y}_2]}$ into $\{\mathcal{V}_{[\text{MASK}]_{m+1}}, \mathcal{V}_{[\text{MASK}]_{m+2}}, \ldots, \mathcal{V}_{[\text{MASK}]_{m+n}}\}$ according to the token's position in the answer. The mapping function $\phi_j(\text{y})$ is to map answer y to the set of label words $\mathcal{V}_{[\text{MASK}]_j}$ for the j-th masked position $[\text{MASK}]_j$.

Here, we still take the disease "*Angina pectoris*" and category "*description*" as an example. As shown in Figure 5, the template and label word set can be formalized as follows:

$$\begin{aligned}
T(x) &= [\text{MASK}]_1, \ldots, [\text{MASK}]_m\, [\text{P}_1], \ldots, [\text{P}_l] \\
&\quad [\text{MASK}]_{m+1}, \ldots, [\text{MASK}]_{m+n} [\text{P}_{l+1}], \ldots, [\text{P}_{2l}]\; x \\
\mathcal{V}_{[\text{MASK}]_1} &= \{\text{“}Diabetes\text{”}, \text{“}Angina\text{”}, \text{“}Iron\text{”}, \text{“}\ldots\text{”}\} \\
\mathcal{V}_{[\text{MASK}]_2} &= \{\text{“}pectoris\text{”}, \text{“}deficiency\text{”}, \text{“}\ldots\text{”}\} \\
\mathcal{V}_{[\text{MASK}]_3} &= \{\text{“}anemia\text{”}, \text{“}\ldots\text{”}\} \\
&\ldots \\
\mathcal{V}_{[\text{MASK}]_{m+1}} &= \{\text{“}cause\text{”}, \text{“}description\text{”}, \text{“}checking\text{”}, \text{“}\ldots\text{”}\} \\
\mathcal{V}_{[\text{MASK}]_{m+2}} &= \{\text{“}item\text{”}, \text{“}\ldots\text{”}\} \\
&\ldots
\end{aligned} \tag{8}$$

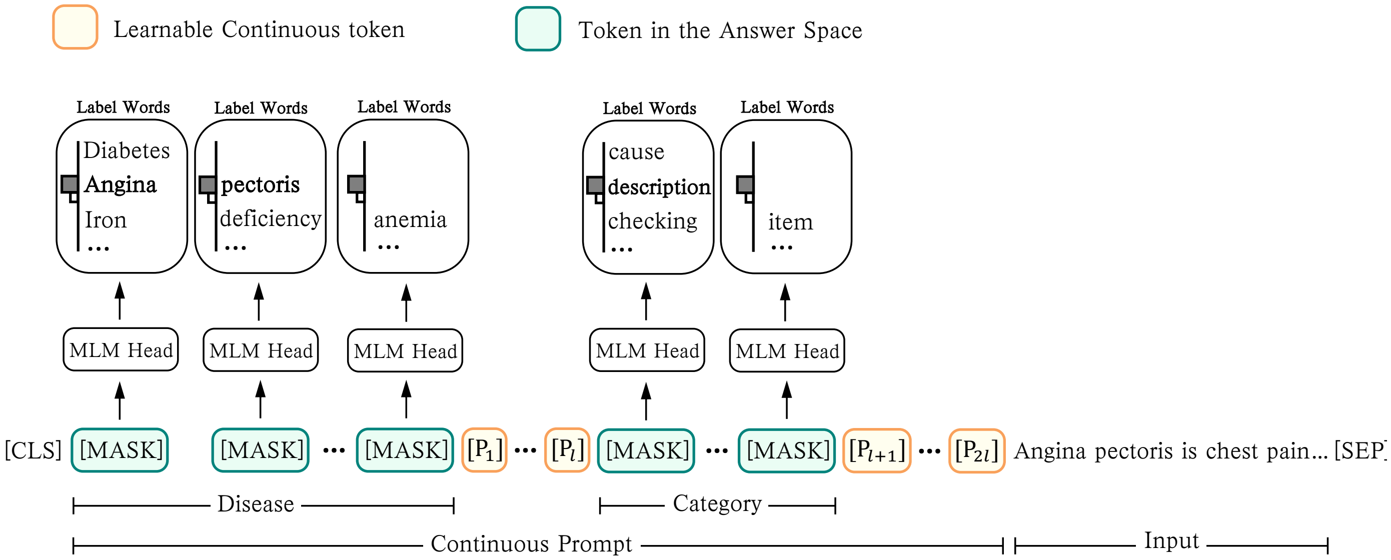


Figure 5. Illustration of KLAPrompt in the Medical Field. We use auxiliary virtual tokens $[\text{P}_1], [\text{P}_2], \ldots, [\text{P}_{2l}]$ to replace natural language words and design the continuous prompt and for DCP dataset.

As the template may contain multiple [MASK] tokens, we must consider all masked positions to make predictions, i.e.,

$$p(y|x) = \prod_{j=1}^{m+n} p([\text{MASK}]_j = \phi_j(\text{y})|T(x)) \tag{9}$$

where $m+n$ is the number of masked positions in $T(x)$, and $\phi_j(\text{y})$ is to map the answer y to the set of label words $\mathcal{V}_{[\text{MASK}]_j}$ for the j-th masked position $[\text{MASK}]_j$. The loss function is given by

$$\begin{aligned}
\mathcal{L} &= -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \log p(y|x) \\
&= -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \log \prod_{j=1}^{m+n} p([\text{MASK}]_j = \phi_j(\text{y})|T(x))
\end{aligned} \tag{10}$$

4. Experiments

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

4.1. Datasets

4.1.1. Open Domain

STS-B. This Chinese version of the dataset (https://github.com/pluto-junzeng/CNSD accessed on 15 April 2023) is translated from the original English dataset STS-B [39] and partially manually revised. Semantic textual similarity (STS) measures the meaning similarity of sentences.

Book Review. The Book Review dataset [40] is collected from Douban, a Chinese online review website that provides information about books, movies, and music. It is a one-sentence text classification dataset.

XNLI. In our experiment, only the Chinese part of the Cross-Language Natural Language Inference (XNLI) [41] dataset 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 [40] is a sentiment analysis dataset that contains 12,000 hotel reviews. In total, 6000 reviews are positive, and the other 6000 reviews are negative.

IFLYTEK. The IFLYTEK [42] dataset has more than 17,000 long texts containing application descriptions, including various application topics related to daily life, with a total of 119 categories.

The datasets above contain 8.05 K, 40.0 K, 40.0 K, 12.0 K, and 17.3 K samples, respectively. We follow the evaluation metrics and settings used in [40,42].

4.1.2. Biomedical Domain

We utilize five biomedical datasets over three different tasks to evaluate our method. The CMeIE, CHIP-CDN, CHIP-CTC, KUAKE-QQR, and KUAKE-QTR datasets are proposed in [43].

CMeIE. The task of Chinese Medical Information Extraction (CMeIE) is to identify medical entities from complex medical text data and determine the relationships between such medical entities. This dataset includes 518 pediatric diseases and 109 common diseases, providing rich resources for medical-related natural language processing research. CMeIE contains nearly 75,000 triplet data, 28,000 sentences about disease descriptions, and 53 different schemas.

CHIP-CDN. The goal of the Clinical Diagnosis Normalization (CHIP-CDN) task is to find a unified and comparable standard for different terms. Based on standardized terminology, researchers can effectively conduct statistical analysis and obtain more accurate results. CHIP-CDN is a semantic matching task. It provides 2500 standardized surgical data to improve task performance.

CHIP-CTC. Recruiting clinical trial subjects requires careful comparison and strict screening. Due to the complex and time-consuming recruitment process, many clinical trials cannot proceed as planned, and a large number of participants withdraw in the middle of the experiment, which seriously affects the effectiveness of the experiment. The Clinical Trial Criterion (CHIP-CTC) task is based on clinical trial screening standards to classify subjects in clinical trials.

KUAKE-QQR. The Query-Query Relevance (KUAKE-QQR) task mainly evaluates the degree of matching between two query topics. It aims to determine whether Query-A and Query-B have undergone translation and the degree of translation. Calculating the correlation between two query terms is an important task that can optimize the search quality of long-tail queries.

KUAKE-QTR. The Query-Title Relevance (KUAKE-QTR) task mainly evaluates the degree of matching between the query topic and the title topic, which is related to the accuracy of the search results. This task requires determining whether the Query topic and Title topic are consistent.

The datasets above contain 22.4 K, 18.2 K, 40.6 K, 18.2 K, and 32.6 K samples, respectively. We follow the evaluation metrics used in [43].

4.2. Implementation Details

KLAPrompt is based on pre-trained language models. In the open domain, we choose BERT [1], RoBERTa [3], and MacBERT [44] as our basic models. In the biomedical domain, we choose BERT [1], MacBERT [44], MC-BERT [36], MedBERT (https://github.com/trueto/medbert accessed on 15 April 2023), and SMedBERT [37] as our baselines. For all these models, the number of layers is 12, the hidden size is 768, the number of heads is 12, and it contains 110 M parameters. These models are optimized with the Adam optimizer [45], with the initial learning rate of 1×10^{-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 24 GB card.

4.3. Results on Open Domain

The experimental results on the development set of five Chinese natural language processing datasets are presented in Table 2. We show each original model and the model trained with the KLAPrompt method (e.g., "BERT" and "BERT + KLAPrompt"). We find that all pre-trained language models trained with the KLAPrompt method have achieved significant improvements compared to the original PLMs. For the STS-B, Book Review, and XNLI datasets, "RoBERTa + KLAPrompt" increases the final results by 2.14%, 2.04%, and 2.32%. Moreover, for the IFLYTEK dataset, the method still raises the accuracy by more than 1%. This superior performance proves that infusing external semantic knowledge via the KLAPrompt approach can empower the widely adopted pre-trained language models.

Table 2. Experimental results of baseline methods and our method on five datasets (Acc.%). "+ KLAPrompt" indicates that we train the PLMs with the KLAPrompt method via semantic knowledge infusion training before fine-tuning.

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** ↑

In our proposed KLAPrompt, five components may affect the performance: the discrete prompt, continuous prompt, discrete answer, continuous answer, and sentence similarity. To explore such effects, we conduct an ablation experiment using the XNLI dataset. The experimental results are presented in Table 3.

We first compare BERT with BERT + WSP † to showcase the advantages of external semantic knowledge in the WSP dataset. BERT + WSP † is trained on the WSP dataset with its original masked language model (MLM), and it does not use the KLAPrompt method. The experimental results demonstrate that introducing semantic information from the Xinhua Dictionary can consistently improve language modeling and downstream tasks.

Then, we compare "BERT + WSP †" with "BERT + Discrete Prompt †" and "BERT + Continuous Prompt †". In order to control the variables for comparison, in this group of experiments, both "BERT + Discrete Prompt †" and "BERT + Continuous Prompt †" use BERT's original masked language model (MLM) without using long-answer strategies. We find that both discrete and continuous prompts can improve the performance of the model.

Table 3. Ablation study on XNLI dataset (Acc.%). "+ WSP" indicates that we train BERT on the WSP dataset without the KLAPrompt approach. † indicates that we train these models on the WSP dataset before fine-tuning.

Models	**XNLI**
BERT	76.8
BERT + WSP †	77.34 (+0.54)
BERT + Discrete Prompt †	77.52 (+0.72)
BERT + Continuous Prompt †	77.72 (+0.92)
BERT + Continuous Prompt + Discrete Answer †	**78.61 (+1.81)**
BERT + Continuous Prompt + Continuous Answer †	78.28 (+1.48)
BERT + Continuous Prompt + Sentence Similarity †	78.41 (+1.61)

We further compare "BERT + Continuous Prompt + Discrete Answer †", "BERT + Continuous Prompt + Continuous Answer †", and "BERT + Continuous Prompt + Sentence Similarity †". Both the discrete answer and continuous answer use answer space partitioning strategies. Among these three long-answer strategies, the discrete answer with the answer space partitioning strategy results in the best performance.

In Table 4, we offer a detailed comparison between different discrete prompts and continuous prompts. We find that continuous prompts outperform discrete prompts in every dataset. However, it is not possible to consider all possible discrete prompts because the manually crafted prompts are complicated and infinite. Thus, in most cases, we can directly utilize the continuous prompts in prompt learning.

The hyper-parameter l is the number of virtual tokens in continuous prompts. To explore its impact on the performance of our prompt learning methods, we test them with different values of hyper-parameter $l = \{1, 2, 3, 4, 5\}$. As shown in Table 4 and Figure 6, different tasks reach their best performance with different values of hyper-parameter l. Thus, the hyper-parameter l needs to be tuned according to downstream NLP task.

We also investigate the consistency of the improvements with different percentages of downstream training data. The experiment results in Figure 7 illustrate that the improvement is more obvious when the amount of data is smaller. In other words, prompt learning with semantic knowledge can benefit data-scarce downstream tasks because, when the training data are limited, the task depends on the pre-trained language model and the additional semantic knowledge.

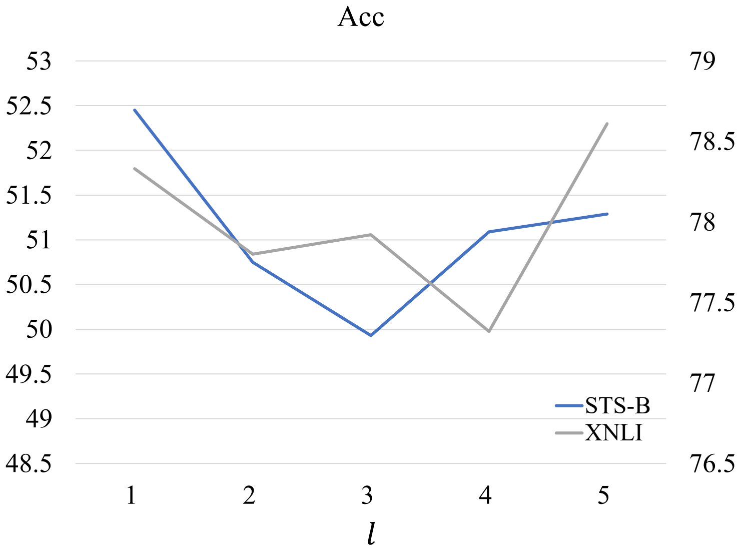


Figure 6. Performance of prompt learning with continuous prompts method with respect to different values of hyper-parameter l.

Table 4. Comparison with discrete prompts and continuous prompts (Acc.%). In this table, we use discrete answers.

Prompt Type	Template T(x)	STS-B	Book Review	XNLI	Chnsenticorp	IFLYTEK
Discrete	The meaning of [C] is [Y]. [X]	**52.38**	86.57	**77.8**	94.49	60.43
	The meaning of "[C]" is [Y]. [X]	51.63	86.05	76.64	94.65	60.4
	Among them, the meaning of [C] is [Y]. [X]	49.12	**86.92**	77.04	**94.74**	**60.44**
	In this sentence, the meaning of [C] is [Y]. [X]	51.36	86.71	77.68	94.41	60.25
Continuous	[C] $[P_1], \ldots, [P_l]$ [Y]. [X], l = 1	**52.45**	88.58	78.33	**94.82**	60.14
	[C] $[P_1], \ldots, [P_l]$ [Y]. [X], l = 2	50.75	**88.63**	77.8	94.65	60.25
	[C] $[P_1], \ldots, [P_l]$ [Y]. [X], l = 3	49.93	88.37	77.92	94.57	59.87
	[C] $[P_1], \ldots, [P_l]$ [Y]. [X], l = 4	51.09	88.52	77.32	94.33	**60.56**
	[C] $[P_1], \ldots, [P_l]$ [Y]. [X], l = 5	51.29	88.39	**78.61**	94.57	60.55

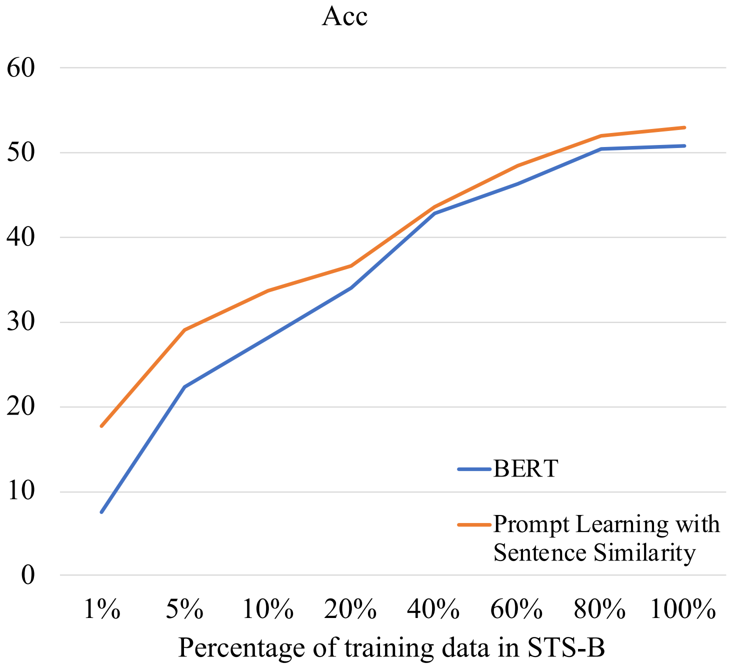


Figure 7. Performance of BERT and prompt learning with sentence similarity method with different amounts of training data.

5. Discussion

In view of the comparison results on five open-domain datasets, namely STS-B (https://github.com/pluto-junzeng/CNSD accessed on 15 April 2023), Book Review [40], XNLI [41], Chnsenticorp [40], and IFLYTEK [42], it is clear that our KLAPrompt method can achieve superior performance in open-domain downstream tasks based on the strength of the fine-grained semantic knowledge. In order to further investigate the effectiveness of the KLAPrompt method in the medical field, we conduct extensive experiments on five biomedical domain datasets: CMeIE, CHIP-CDN, CHIP-CTC, KUAKE-QQR, and KUAKE-QTR [43].

Results on Biomedical Domain

The experimental results on the development set of five Chinese biomedical datasets are presented in Table 5. We find that all pre-trained language models trained with the KLAPrompt method have achieved significant improvements compared to the original PLMs. For the MacBERT, MC-BERT, and SMedBERT models trained with the KLAPrompt method, they increase the final results on the CMeIE dataset by 2.75%, 2.73%, and 2.12%. Moreover, for most datasets, the method still can increase the results by more than 1%. This superior performance proves the effectiveness of the KLAPrompt method in the medical field.

In KLAPrompt, other two components may affect the performance: disease prediction and category prediction. To explore such effects, we conduct an ablation experiment using the CMeIE dataset. We first compare "SMedBERT" with "SMedBERT + DCP †" to showcase the advantages of external disease knowledge in the DCP dataset. "SMedBERT + DCP †" is trained on the DCP dataset with its original masked language model (MLM), and it does not use the KLAPrompt method. The experimental results demonstrate that introducing disease information from MedicalKG can consistently improve language modeling and downstream tasks.

Then, we explore the effects of the two components. "- Disease Prediction" indicates that the disease prediction component is removed during training. As shown in Table 6, both components can improve the performance on this dataset. In addition, we observe the worst result when we remove the disease prediction, which shows that disease prediction is more effective than category prediction.

Table 5. Experimental results of baseline methods and our method on five datasets. "+ KLAPrompt" indicates that we train the PLMs with the KLAPrompt method via disease knowledge infusion training before fine-tuning.

Tasks	**Relation Extraction**	**Text Classification**	**Sentence Similarity Estimation**		
Datasets	**CMeIE**	**CHIP-CDN**	**CHIP-CTC**	**KUAKE-QQR**	**KUAKE-QTR**
Metrics (%)	**F1**	**F1**	**F1**	**Accuracy**	**Accuracy**
BERT	71.04	75.12	83.3	81.49	66.29
BERT + KLAPrompt	72.12 ↑	76.28 ↑	84.12 ↑	82.99 ↑	67.49 ↑
MacBERT	69.89	76.01	83.19	82.68	66.53
MacBERT + KLAPrompt	72.64 ↑	77.23 ↑	83.9 ↑	82.99 ↑	**67.77** ↑
MC-BERT	69.08	76.51	82.2	82.36	66.3
MC-BERT + KLAPrompt	71.81 ↑	**77.44** ↑	83.58 ↑	83.05 ↑	67.54 ↑
MedBERT	71.63	76.85	82.74	80.49	66.13
MedBERT + KLAPrompt	72.35 ↑	78.53 ↑	84.12 ↑	82.05 ↑	67.2 ↑
SMedBERT	70.53	75.49	82.56	82.36	65.53
SMedBERT + KLAPrompt	**72.65** ↑	76.73 ↑	**84.25** ↑	**83.55** ↑	66.98 ↑

Table 6. Ablation study on the CMeIE dataset (F1%). † indicates that we train these models on the DCP dataset before fine-tuning.

Variants	**F1**
SMedBERT	70.53
SMedBERT + DCP †	70.97
- Disease Prediction †	71.88
- Category Prediction †	72.33
SMedBERT + KPL †	**72.65**

6. Conclusions

In this work, we propose a long-answer prompt learning method (KLAPrompt) with three different long-answer strategies to introduce fine-grained semantic knowledge from the Xinhua Dictionary. According to the different forms of answers, they can be divided into three strategies: discrete answers, continuous answers, and sentence similarity. In the discrete answer strategy, we split the answer space into several answer subspaces according to the token's position in the long answer and predict each word of the sense independently. In the continuous answer strategy, we use virtual tokens to replace the natural language in discrete answers. These virtual tokens can be embedded in a continuous space and optimized through gradient descent. In the sentence similarity strategy, we average the embeddings of the masked part in the MLM output, calculate the cosine similarity between this and the sentence embedding of the original long answer, and then maximize the similarity during the training procedure. Furthermore, we explore the effectiveness of the KLAPrompt method in the medical field. We apply the KLAPrompt method to introduce fine-grained disease knowledge from MedicalKG into pre-trained language models. Furthermore, we collect a word sense prediction dataset (WSP) based on the Xinhua Dictionary and a disease and category prediction dataset (DCP) based on MedicakKG.

Experimental results on five open-domain datasets demonstrate that all pre-trained language models trained with the KLAPrompt method have achieved significant improvements compared to the original PLMs. The superior performance proves that the infusion of external semantic knowledge from the Xinhua Dictionary can empower the widely adopted pre-trained language models. We also find that both discrete and continuous prompts can improve the performance of the model, and continuous prompts outperform discrete prompts in all datasets. Thus, in most cases, we can directly utilize continuous prompts in prompt learning. Among the three long-answer strategies, the discrete answer strategy is the best method to integrate structured semantic knowledge. Moreover, we investigate the consistency of the improvements with different percentages of downstream training data. The experimental results illustrate that the improvement is more obvious when the amount of data is smaller. In other words, prompt learning with semantic knowledge can benefit data-scarce downstream tasks.

Additionally, we conduct comprehensive experiments on five health-related datasets to explore the effectiveness of our KLAPrompt method in the medical field. Extensive experiments verify that pre-trained language models with the KLAPrompt method can also achieve superior performance based on the strength of the disease knowledge in MedicalKG. Then, we explore the effects of two components: disease prediction and category prediction. We observe the worst result when we remove the disease prediction, which shows that disease prediction is more effective than category prediction.

In our future work, we will add some virtual tokens on the left of the predicted word [C], rather than only on one side. We will also infuse common-sense information, domain-specific information, and knowledge graphs into the pre-trained language models. In the event that incorrect semantic content is provided, the outcome of the pre-trained learning might also be misleading, and this may significantly influence the outcome of the language processing. Thus, we will delve deeper into this problem in our future work.

Author Contributions: H.-T.Z. and Z.X. were responsible for the overall design of the study. H.-T.Z. and Z.X. performed the experiments and drafted the manuscript. Z.X., H.-T.Z., W.L., D.H., B.W. and H.-G.K. revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by 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), the Major Key Project of PCL for Experiments and Applications (PCL2021A06), and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (HW2021008).

Data Availability Statement: Publicly available datasets were generated in this study. This data can be found here: https://github.com/Xie-Zuotong/WSP and https://github.com/Xie-Zuotong/DCP.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (Long and Short Papers), pp. 4171–4186.
2. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5753–5763.
3. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
4. Zellers, R.; Bisk, Y.; Schwartz, R.; Choi, Y. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 93–104.
5. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.* **2020**, *63*, 1872–1897. [CrossRef]
6. Han, X.; Zhang, Z.; Ding, N.; Gu, Y.; Liu, X.; Huo, Y.; Qiu, J.; Yao, Y.; Zhang, A.; Zhang, L.; et al. Pre-trained models: Past, present and future. *AI Open* **2021**, *2*, 225–250. [CrossRef]
7. Feng, X.; Feng, X.; Qin, B.; Feng, Z.; Liu, T. Improving Low Resource Named Entity Recognition using Cross-lingual Knowledge Transfer. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; Volume 1, pp. 4071–4077.
8. Zhang, S.; Zhang, Y.; Chen, Y.; Wu, D.; Xu, J.; Liu, J. Exploiting Morpheme and Cross-lingual Knowledge to Enhance Mongolian Named Entity Recognition. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2022**, *21*, 94. [CrossRef]
9. Li, Z.; Ding, N.; Liu, Z.; Zheng, H.; Shen, Y. Chinese relation extraction with multi-grained information and external linguistic knowledge. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4377–4386.
10. Alt, C.; Gabryszak, A.; Hennig, L. Probing Linguistic Features of Sentence-Level Representations in Neural Relation Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 1534–1545.
11. Aharoni, R.; Goldberg, Y. Towards String-To-Tree Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 2: Short Papers, pp. 132–140.
12. Yu, Z.; Xian, Y.; Yu, Z.; Huang, Y.; Guo, J. Linguistic feature template integration for Chinese-Vietnamese neural machine translation. *Front. Comput. Sci.* **2022**, *16*, 163344. [CrossRef]
13. Yu, Z.; Yu, Z.; Xian, Y.; Huang, Y.; Guo, J. Improving Chinese-Vietnamese Neural Machine Translation with Linguistic Differences. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2022**, *21*, 1–12. [CrossRef]
14. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT understands, too. *arXiv* **2021**, arXiv:2103.10385.
15. Cui, L.; Wu, Y.; Liu, J.; Yang, S.; Zhang, Y. Template-Based Named Entity Recognition Using BART. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 1835–1845.
16. Mahabadi, R.K.; Zettlemoyer, L.; Henderson, J.; Saeidi, M.; Mathias, L.; Stoyanov, V.; Yazdani, M. 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), Dublin, Ireland, 22–27 May 2022; Volume 1: Long Papers, pp. 3638–3652.
17. Cui, G.; Hu, S.; Ding, N.; Huang, L.; Liu, Z. Prototypical Verbalizer for Prompt-based Few-shot Tuning. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Volume 1: Long Papers, pp. 7014–7024.
18. Miller, G.A. WordNet: A lexical database for English. *Commun. Acm* **1995**, *38*, 39–41. [CrossRef]
19. Dong, Z.; Dong, Q. HowNet-a hybrid language and knowledge resource. In Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering, Beijing, China, 26–29 October 2003; pp. 820–824.
20. Bevilacqua, M.; Pasini, T.; Raganato, A.; Navigli, R. Recent trends in word sense disambiguation: A survey. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Montreal, QC, Canada, 19–26 August 2021.
21. Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; McCallum, A. Linguistically-Informed Self-Attention for Semantic Role Labeling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 5027–5038.
22. Kong, D.; Li, X.; Wang, S.; Li, J.; Yin, B. Learning visual-and-semantic knowledge embedding for zero-shot image classification. *Appl. Intell.* **2022**, *53*, 2250–2264. [CrossRef]
23. Kiefer, S. CaSE: Explaining text classifications by fusion of local surrogate explanation models with contextual and semantic knowledge. *Inf. Fusion* **2022**, *77*, 184–195. [CrossRef]
24. Grand, G.; Blank, I.A.; Pereira, F.; Fedorenko, E. Semantic projection recovers rich human knowledge of multiple object features from word embeddings. *Nat. Hum. Behav.* **2022**, *6*, 975–987. [CrossRef] [PubMed]
25. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. Ernie: Enhanced representation through knowledge integration. *arXiv* **2019**, arXiv:1904.09223.
26. Peters, M.E.; Neumann, M.; Logan, R.; Schwartz, R.; Joshi, V.; Singh, S.; Smith, N.A. 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), Hong Kong, China, 3–7 November 2019; pp. 43–54.

27. Levine, Y.; Lenz, B.; Dagan, O.; Ram, O.; Padnos, D.; Sharir, O.; Shalev-Shwartz, S.; Shashua, A.; Shoham, Y. SenseBERT: Driving Some Sense into BERT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 4656–4667.
28. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv* **2021**, arXiv:2107.13586.
29. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
30. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
31. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *36*, 1234–1240. [CrossRef]
32. Peng, Y.; Yan, S.; Lu, Z. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In Proceedings of the 18th BioNLP Workshop and Shared Task, Florence, Italy, 1 August 2019; pp. 58–65.
33. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. 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), Hong Kong, China, 3–7 November 2019; pp. 3615–3620.
34. Huang, K.; Altosaar, J.; Ranganath, R. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv* **2019**, arXiv:1904.05342.
35. Gu, Y.; Tinn, R.; Cheng, H.; Lucas, M.; Usuyama, N.; Liu, X.; Naumann, T.; Gao, J.; Poon, H. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthc.* **2021**, *3*, 2. [CrossRef]
36. Zhang, N.; Jia, Q.; Yin, K.; Dong, L.; Gao, F.; Hua, N. Conceptualized representation learning for chinese biomedical text mining. *arXiv* **2020**, arXiv:2008.10813.
37. Zhang, T.; Cai, Z.; Wang, C.; Qiu, M.; Yang, B.; He, X. SMedBERT: A Knowledge-Enhanced Pre-trained Language Model with Structured Semantics for Medical Text Mining. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; Volume 1: Long Papers, pp. 5882–5893.
38. Hambardzumyan, K.; Khachatrian, H.; May, J. WARP: Word-level Adversarial ReProgramming. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; Volume 1: Long Papers, pp. 4921–4933.
39. Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; Specia, L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv* **2017**, arXiv:1708.00055.
40. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-bert: Enabling language representation with knowledge graph. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 2901–2908.
41. Conneau, A.; Rinott, R.; Lample, G.; Schwenk, H.; Stoyanov, V.; Williams, A.; Bowman, S.R. XNLI: Evaluating cross-lingual sentence representations. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, Brussels, Belgium, 31 October–4 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 2475–2485.
42. Xu, L.; Hu, H.; Zhang, X.; Li, L.; Cao, C.; Li, Y.; Xu, Y.; Sun, K.; Yu, D.; Yu, C.; et al. CLUE: A Chinese Language Understanding Evaluation Benchmark. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 4762–4772.
43. Zhang, N.; Chen, M.; Bi, Z.; Liang, X.; Li, L.; Shang, X.; Yin, K.; Tan, C.; Xu, J.; Huang, F.; et al. CBLUE: A Chinese Biomedical Language Understanding Evaluation Benchmark. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Volume 1: Long Papers, pp. 7888–7915.
44. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Revisiting Pre-Trained Models for Chinese Natural Language Processing. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 657–668.
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.