Hindawi
Security and Communication Networks
Volume 2021, Article ID 7115167, 8 pages
https://doi.org/10.1155/2021/7115167

Research Article

A General Method for Transferring Explicit Knowledge into Language Model Pretraining

Ruiqing Yan,[1] **Lanchang Sun**,[2] **Fang Wang**,[1] **and Xiaoming Zhang**[1]

[1]*College of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China*
[2]*School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Fang Wang; fangwang@bipt.edu.cn

Received 22 May 2021; Accepted 9 September 2021; Published 8 October 2021

Academic Editor: Feiran Huang

Recently, pretrained language models, such as Bert and XLNet, have rapidly advanced the state of the art on many NLP tasks. They can model implicit semantic information between words in the text. However, it is solely at the token level without considering the background knowledge. Intuitively, background knowledge influences the efficacy of text understanding. Inspired by this, we focus on improving model pretraining by leveraging external knowledge. Different from recent research that optimizes pretraining models by knowledge masking strategies, we propose a simple but general method to transfer explicit knowledge with pretraining. To be specific, we first match knowledge facts from a knowledge base (KB) and then add a knowledge injunction layer to a transformer directly without changing its architecture. This study seeks to find the direct impact of explicit knowledge on model pretraining. We conduct experiments on 7 datasets using 5 knowledge bases in different downstream tasks. Our investigation reveals promising results in all the tasks. The experiment also verifies that domain-specific knowledge is superior to open-domain knowledge in domain-specific task, and different knowledge bases have different performances in different tasks.

1. Introduction

Recently, substantial work has shown that pretrained models [1–4] can learn language representations over large-scale text corpora, which are beneficial for many downstream NLP tasks. For example, XLNet [5] obtains new state-of-the-art results on twelve NLP tasks including reading comprehension, question answering, and text classification. Researchers [6–8] ascertain that pretraining allows models to learn syntactic and semantic information of language that is then transferred to other tasks. Most of the existing works model the representations by predicting the missing word only through the contexts. It is solely at the word token level [9] without considering the background knowledge in the text.

Background knowledge [10] comprises all of the world knowledge that the reader brings to the task of reading. This can include episodic (events), declarative (facts), and procedural (how-to) knowledge as well as related vocabulary. Background knowledge influences the efficacy of understanding. It has been considered an indispensable part of language understanding [11–13]. For instance, one major step in improving reading is to improve prior knowledge of the topics being read [14]. We argue that background knowledge can lead to better language understanding. For example, given the sentence "*Xiaomi was officially listed on the main board of HKEx*," the background knowledge may include *Xiaomi* is a *science and technology company*, *HKEx* refers to *Hong Kong Exchanges and Clearing Limited*, and *main board* is an *economic term*. Knowing these knowledge facts can help us better understand the word sense and the sentence topic.

Explicit knowledge is knowledge that can be readily articulated, codified, stored, and accessed (https://enẇikipedia□rg/wiki/Explicit_knowledge), such as Freebase [15] and DBpedia [16]. For example, "*Xiaomi* is a *science and technology company*" is kind of typical explicit knowledge. It could be well stored in knowledge bases, represented in the form of SPO (subject, predicate, and object) triplet [17], where subject and object are entities and

predicate is a relation between those entities. This paper seeks to find the impact of explicit knowledge on transformer pretraining.

How to inject explicit knowledge from external sources into the transformer-based language models has gradually become a research hotspot [18]. There are some improved models based on Bert [19–21] or GPT [22], which prove that injecting extra knowledge information can significantly enhance original models. The difference between these methods lies in the different ways of knowledge injection. For example, ERNIE [19] refines the transformer architecture by using entity-level masking and phrase-level masking. KALM [22] signals the existence of entities to the input of the transformer in pretraining and adds an entity prediction task in the output.

In this paper, we propose a simple but general method for transferring knowledge into LM pretraining without changing the model architecture. Taking XLNet as a running example, given a sentence, we first use a dictionary look up method to map its knowledge facts. A knowledge injection layer is then designed to combine external knowledge with the original sentence, in a way that is close to natural language and accepted by XLNet without losing the structure information. Finally, we take the output of the knowledge injection layer directly as the input for XLNet and design a three-stage training method to save training time and hardware cost. To investigate the impact of explicit knowledge, we leverage open-domain and domain-specific knowledge to combine with XLNet and test their performances on various NLP tasks.

Our contributions in this paper are threefold: (1) proposal of a simple but general knowledge transferring method for language model pretraining, (2) proposal of K-XLNet for implementation of the proposed method on XLNet, and (3) empirical verification of the effectiveness of K-XLNet on various downstream tasks.

The rest of the paper is organized as follows. Section 2 summarizes related work. In Section 3, we elaborate on our knowledge transferring method taking XLNet as a running example. Section 4 reports experimental results, and finally Section 5 concludes the whole paper.

2. Related Work

In recent years, with the rapid development of deep learning, the pretraining technology [1–3, 23, 24] in the field of natural language processing has made great progress. Many efforts [4, 5] are devoted to pretraining language representation models for learning language representations over large-scale corpus and then utilizing the representations for downstream NLP tasks such as question answering [25] and text classification [26].

Petroni et al. [8] discovered that language models embed some facts and relationships in their weights during pretraining. This can help explain the performance of these models in semantic tasks [27, 28]. However, relaying only on the text corpus, they have some tendency to hallucinate knowledge, whether through bias or incorrect knowledge in the training data [18]. Moreover, they [22] are still far from ready to serve as an "unsupervised multitask learner." There are still gaps between model pretraining and task-specific fine-tuning [29]. Pretraining models usually learn universal language representation from general-purpose large-scale text corpora, but lack domain or task-specific knowledge. This leads to the need of huge efforts for task-specific fine-tuning or overparameterization [30].

Recent work shows that combining with knowledge is a promising way to improve language models. The knowledge bases such as WordNet [31], Freebase [15], DBpedia [16], and ConceptNet [32] contain a large amount of reliable knowledge information. By leveraging DBpedia, K-Bert [21] have revealed promising results in knowledge-driven applications such as named entity recognition, entity typing, and relation extraction. Based on Bert and improved by refining the transformer architecture with Baidu Baike (Baidu Baike is a wiki-like online Chinese encyclopedia), ERNIE [19] and ERNIE1 (THU) [20] achieved good results on five Chinese natural language processing tasks including natural language inference, semantic similarity, named entity recognition, sentiment analysis, and question answering. Based on GTP2.0 [33], KALM [22] significantly improved downstream tasks such as zero-shot question answering by adding entity signals to the input of the transformer and an entity prediction task to the output. KEPLE [34] incorporated factual knowledge into language representations by jointly training with knowledge embedding and the masked language modeling objectives. This makes it also work well as an inductive knowledge embedding model on knowledge graph link prediction.

Our work is similar to AMS [35] and K-Bert [21], in which knowledge triplets are aligned with textual sentences. However, different from them, we focus on finding the direct impact of explicit knowledge on pretraining. Instead of changing the transformer architecture, we take sentences and the related knowledge facts as input into a knowledge injector. The injector is designed to combine the knowledge with original text and generate knowledge enriched output as the input of the transformer. A three-step training method is proposed to combine domain knowledge and pretrained model flexibly. Taking XLNet as a running example, we investigate the impact of open-domain and domain-specific knowledge on pretraining in various downstream NLP tasks.

3. Methodology

We propose a simple but general method for transferring knowledge into language model pretraining. In this paper, we take XLNet as a running example. Figure 1 shows the overall framework. We can see that the proposed method does not change the original architecture of XLNet. A knowledge injunction layer is designed and connected to Transformer-XL. We elaborate on the proposed method in the following three sections.

3.1. Knowledge Matching. For a given sentence, how to effectively match its related knowledge is the primary problem we need to solve. We take SPO triplets in knowledge base

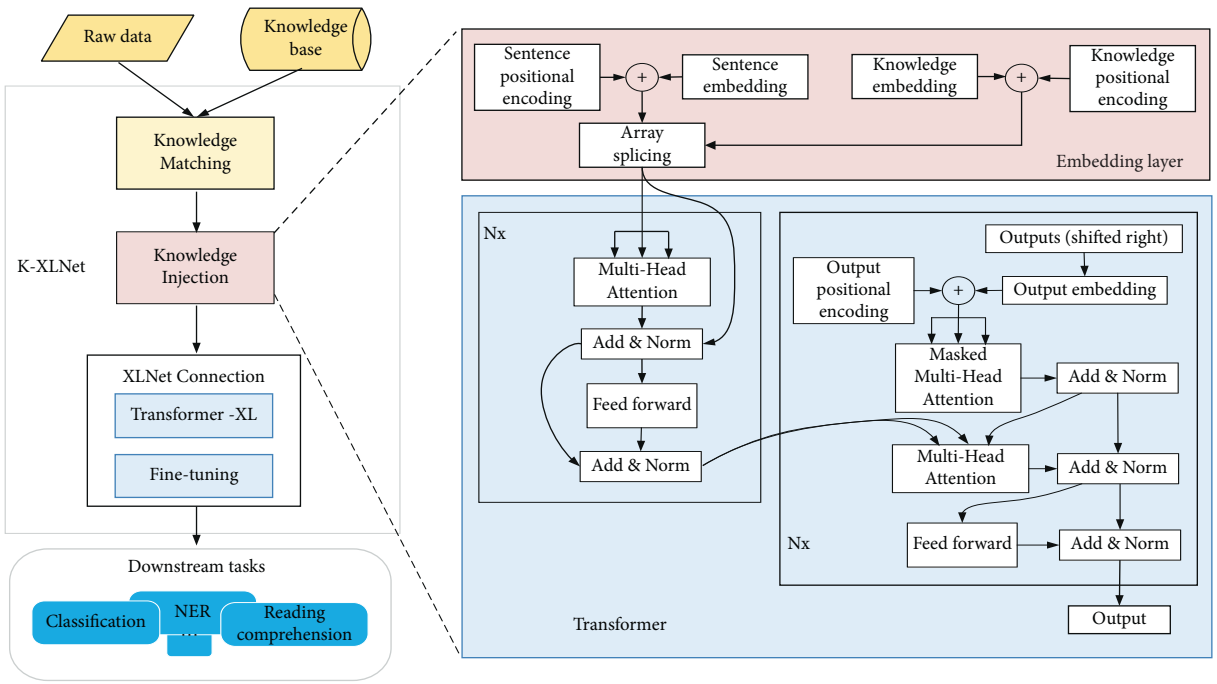

Figure 1: The overall framework of K-XLNet.

(e.g., DBpedia) as the source of knowledge facts. A term lexicon is first built by leveraging all the subjects in knowledge base. Then, we identify subject terms in the given sentence by means of a lexicon lookup. During the parsing, if one term is a substring of another term (e.g., New York and New York Times), we choose the longest term as the subject term. We finally get the related SPO triplets through a frequency-based dictionary lookup. If there are multiple matched terms in knowledge base, we choose the one with highest frequency. The subject term matching is also known in natural language processing as entity linking [36]. Instead of using a more sophisticated entity linker [37], the dictionary lookup is more efficient for large-scale text corpus than entity linking methods, and using a highly tuned entity linker may propagate its own biases into the transformer.

3.2. Knowledge Injection. We aim to inject the structured and confident knowledge information in KBs into that of the high-coverage, contextual information found in language models. This injection would permit the model to incorporate some of the information found in the KB to improve its performance in many downstream tasks. However, it is a challenge to fuse lexical, syntactic, and knowledge information in the pretraining procedure for language representation [20].

Instead of designing a special pretraining objective for the fusion, we aim to integrate knowledge naturally into the original text in a way that conforms to the grammatical rules of natural language. For example, given the sentence "*Xiaomi listed in Hong Kong*" and a knowledge fact (Xiaomi, is_a, science and technology company), generate a knowledge-enriched sentence like "*Xiaomi, a science and technology company, listed in Hong Kong.*" By this way, we let the pretraining model to use clues in the knowledge-enriched text to learn word representation that better reflect how language conveys knowledge.

Therefore, we treat the knowledge injection problem like a machine translation problem and design a knowledge injector with the structure similar to the transformer used in the field of natural language translation [38], as the knowledge injection layer shown in Figure 1. It mainly consists of two modules, i.e., embedding layer and transformer. For the input sentence and knowledge facts, embedding layer first converts them together with positional information into embedding representation and then feeds into the transformer for knowledge combining.

3.2.1. Embedding Layer. The function of the embedding layer is to convert the given sentence and its related knowledge facts into an embedding representation that can be fed into the following transformer.

In order to express the positional information of the original sentence and the SPO triplet, we splice the original sentence with the matched knowledge triplets and use a two-dimensional array composed of two elements to encode the position of each word. The first element is the sequence number of the word position, called absolute position index. The second element is the sequence number of the word that matches the subject in the original sentence, called relative position index.

Figure 2 shows an embedding example. The encoding of words in the original sentence is composed of two elements with the same value because each word in the original sentence matches itself. For instance, the encode of the first word "Xiaomi" is [1, 1]. For the matched knowledge triplets, each SPO triplet is horizontally spliced into a whole

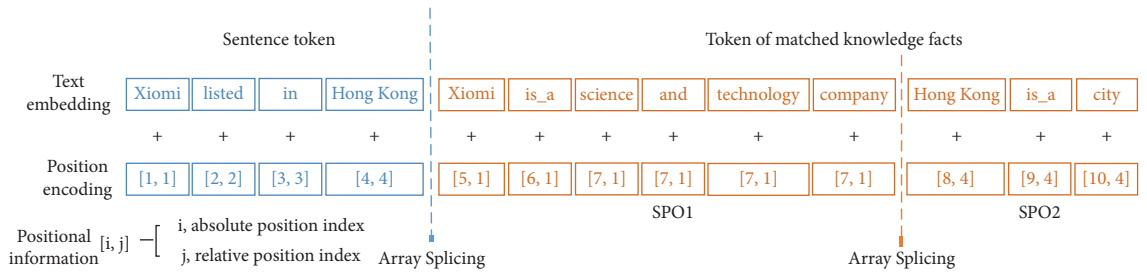

Figure 2: An example of embedding representation.

sequence, and S, P, and O are sequentially indexed. For instance, the encode of the first matched knowledge subject "Xiaomi" is [5, 1]. By this way, the information of the original sentence, matched SPO triplets, and their positions is all preserved. After getting the position code, we use the following formula to normalize the position code:

$$\begin{aligned} \mathrm{PE}_{(\mathrm{pos},2i)} &= \sin\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \\ \mathrm{PE}_{(\mathrm{pos},2i+1)} &= \cos\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right). \end{aligned} \tag{1}$$

3.2.2. Transformer. As shown in Figure 1, the transformer consists of an encoder and a decoder, which has the same structure as the typical transformer [39]. It takes advantage of the positional embedding as a mechanism to encode order within a knowledge-enriched sentence. The encoder stacks 6 identical layers, in which each of them uses the multihead attention and a 2-sublayer feed-forward network, coupled with layer normalization and residual connection. The multihead attention mechanism computes the softmax distribution for each word within a sentence, including the word itself, as shown in the following equation:

$$\mathrm{attention}\,(Q, K, V) = \mathrm{soft\,max}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2}$$

The input consists of queries Q and keys K of dimension d_k and values V of dimension d_v. The queries, keys, and values are linearly projected h times, to allow the model to jointly attend to information from different representations. On top of the multihead attention, there is a feed-forward network that consists of two layers with ReLU activation in between. Each encoder layer takes as input the output of the previous layer, allowing it to attend to all positions of the previous layer. The decoder has a similar architecture as the encoder, but has two multihead attention sublayers.

We use it to inject the matched knowledge triplets to the original sentence. Regarding the matched knowledge triplets as a different language with the original sentence, we turn the knowledge injection problem into a machine translation problem and aim to translate them into a language that confirms to the natural language grammar. By this way, the knowledge-enriched output can be naturally used as the input of Transformer-XL.

3.3. XLNet Connection. XLNet is an advanced transformer-based language model. We take it as a running example for combining knowledge with language model pretraining. The XLNet connected after the knowledge injection layer does not have a tokenization module, namely, the Transformer-XL. Tokenization and encoding have been performed in the embedding module.

Normally, retraining the pretrained model is necessary for model refinement by leverage knowledge information. However, the cost of retraining is very high, both in terms of time and hardware cost. We propose a simple and general way to resolve this problem, inspired by the mainstream idea of pretraining and fine-tuning. Figure 3 shows the process of training K-XLNet. It mainly has three stages: XLNet pre-training, task-specific fine-tuning, and K-XLNet training for the specific task. The first two stages are consistent with the usual two-stage pretraining model. Instead of pretraining K-XLNet on large-scale general corpus, we train it on specific tasks by leveraging external knowledge. In addition to cost saving, this approach makes it easy to flexibly test the effects of different knowledge bases on different downstream tasks. The following experiments show that this method is effective.

4. Experiment

In this section, we evaluate the performance of K-XLNet through seven downstream tasks, among which one is an English task for a specific domain and six are Chinese tasks for open domain.

4.1. Experiment Setup

4.1.1. Preprocessing. We use the pretrained word2vec (https://github&cdotom/xgli/word2vec-api) for word embedding, which was trained on Google News. To cover the new words from knowledge and datasets of downstream tasks, we retrain it on knowledge SPO triplets and task-specific corpus.

4.1.2. Knowledge Graph. We leverage HowNet (https://www.keenage&cdotom/) and CN-DBpedia (https://kw.fudan.edu.cn/cndbpedia/intro/) for Chinese tasks and DBpedia (https://wiki.dbpedia.org/) and MovieKG (https://www.openkg.cn/dataset/keg-bilingual-moviekg) for English tasks. We do not do any preprocessing for these KGs, since

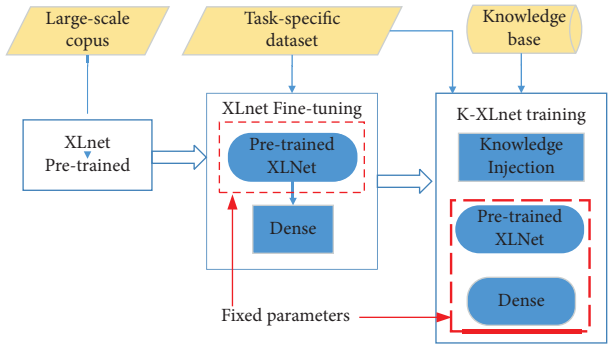

Figure 3: A simple method for training K-XLNet.

only the matched knowledge SPO triplets will be used for training K-XLNet. Detailed information of the KGs is as follows:

(i) HowNet is a large-scale language knowledge base for Chinese vocabulary and concepts [40]. It contains 52576 sets of SPO triplets.

(ii) CN-DBpedia [41] is a large-scale open-domain encyclopedic KG developed by the Knowledge Work Laboratory of Fudan University. It contains 5168865 sets of SPO triplets.

(iii) DBpedia is a large-scale, multilingual knowledge base extracted from Wikipedia [16].

(iv) MovieKG is a bilingual movie knowledge graph constructed by the knowledge engineering laboratory of the Department of Computer Science, Tsinghua University. Unfortunately, the database is offline at present. It includes 23 concepts, 91 attributes, 0.7+ million entities, and 10+ million triplets. Its data sources include LinkedIMDb, Baidu Baike, and Douban.

(v) MedicalKG is a map of Chinese medical knowledge, which contains 13864 sets of SPO triplets.

4.1.3. Baselines. The proposed method is model independent. Therefore, we compare K-XLNet to two released XLNet models (https://github.com/zihangdai/xlnet): XLNet-Base and XLNet-Large. Following XLNet, we design two K-XLNet models as follows:

(i) K-XLNet-Base has 3-layer, 128-hidden, and 2-heads in the knowledge injection layer, with the same Transformer-XL parameters as XLNet-Base

(ii) K-XLNet-Large has 4-layer, 128-hidden, and 3-heads in the knowledge injection layer, with the same Transformer-XL parameters with XLNet-Large

4.2. Domain-Specific Task. We first compare the performance of K-XLNet with the original XLNet on an English domain-specific task, namely, emotion classification for movie reviews.

To be specific, we use the IMDB [42] dataset for this test. It includes 25,000 positive reviews and 25,000 negative reviews. We divided them into three parts: *train*, *dev*, and *test*. We used the *train* part to fine-tune the model and then evaluated its performance on the *dev* and *test* parts. For knowledge injection in K-XLNet, we used MovieKG and DBpedia, respectively. Table 1 shows the experimental results.

It can be seen that K-XLNet is superior to the original XLNet in both parameter settings (base and large). This shows that our approach of knowledge injection to XLNet is effective. In addition, MovieKG performs better than DBpedia, indicating that domain knowledge is preferred for domain-specific tasks.

We further investigate the effect of different SPO triplet (knowledge) amounts in K-XLNet. In this test, we use MovieKG for knowledge injection and set the amount of knowledge triplets to be 1,000, 5,000, 6,000, and 7,000, respectively. The results are shown in Figure 4.

We can see that from 1k to 5k, the performances of K-XLNet models are improving with the increase of knowledge injection. After 5k, the performance tends to be stable or even slightly decreased. This gives us a hint that when using knowledge for model improvement, the more is not the better.

In the following experiments, we set the triplet (knowledge) amount to 5k for all K-XLNet models and compare our K-XLNet to the original XLNet using the *Large* setting, since the *Base* setting has similar performance trend.

4.3. Open-Domain Tasks. We conduct six experiments to evaluate the performance of K-XLNet on open-domain tasks. Specifically, Book_review [21], Shopping [21], and Weibo [21] are single-sentence classification tasks. XNLI [43] and LCQMC [44] are two-sentence classification tasks. MSRA-NER [45] is a Named Entity Recognition (NER) task.

(i) Book_review (https://embeddingġithubio/evaluation) contains 20,000 positive reviews and 20,000 negative reviews collected from Douban.

(ii) Shopping (https://shareẇeiyunċom/5xxYiig) is an online shopping review dataset that contains 40,000 reviews, including 21,111 positive reviews and 18,889 negative reviews.

(iii) Weibo (https://share.weiyun.com/5lEsv0w) is a dataset with emotional annotations from Sina Weibo, including 60,000 positive samples and 60,000 negative samples.

(iv) XNLI (https://github.com/NLPchina/XNLI) is a cross-language language understanding dataset in which each entry contains two sentences and the task is to determine their relation ("Entailment," "Contradict," or "Neutral").

(v) LCQMC (https://github.com/Lizhengo/lcqmc_data) is a large-scale Chinese question matching corpus. The goal of this task is to determine if the two questions have a similar intent.

(vi) MSRA-NER (https://github.com/littleWhiteTre/msra_ner) is a NER dataset published by Microsoft.

Table 1: Results on emotion classification task (accuracy, %)

Method	Dev	Test
XLNet-Base	95.32	94.37
K-XLNet-Base (DBpedia)	95.51	94.88
K-XLNet-Base (MovieKG)	**95.82**	**95.03**
XLNet-Large	96.21	95.13
K-XLNet-Large (DBpedia)	96.74	95.62
K-XLNet-Large (MovieKG)	**96.87**	**95.99**

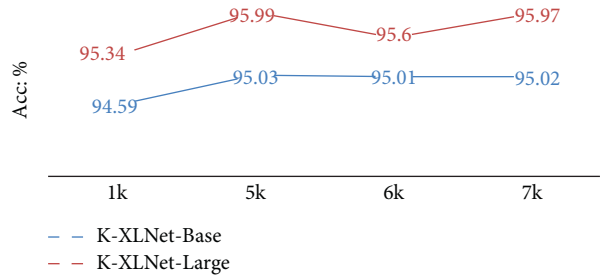

Figure 4: Performance of K-XLNet models injected with different triplet (knowledge) amounts.

Table 2: Results of various models on various open-domain tasks (accuracy, %).

Models/datasets	Book_review		Shopping		Weibo		XNLI		LCQMC		MSRA-NER	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
XLNet	88.71	87.69	96.82	96.73	98.04	97.98	76.87	76.33	88.79	87.25	95.08	94.97
K-XLNet (HowNet)	**88.83**	**88.67**	**97.04**	**97.14**	**98.17**	98.05	**77.18**	**77.13**	**89.02**	**87.37**	96.29	**96.26**
K-XLNet (CN-DBpedia)	88.76	87.71	96.89	96.77	98.12	**98.41**	76.97	76.39	88.87	87.31	**96.31**	96.24

It is used to recognize the named entities in the text, including person names, place names, and organization names\enleadertwodots.

Similarly, the open-domain datasets are split into three parts: *train*, *dev*, and *test*, used for fine-tuning, model selection, and model test, respectively. Table 2 shows the test results of various models in terms of accuracy. We can see that K-XLNet performs better than XLNet consistently on the six open-domain tasks. To be specific, the improvements are significant on NER task, but not on sentence classification tasks. Moreover, the model leveraging HowNet performs better than that using CN-DBpedia on sentence classification tasks, but it is the opposite on the NER task. The above observations show that knowledge injection to XLNet is also effective on open-domain downstream tasks, and it is important to choose appropriate knowledge base according to the specific task.

4.4. Discussion. This work focused on transferring knowledge facts into language model pretraining. The above experiments showed that the proposed method achieved good results in all the downstream tasks. However, the improvements are limited due to the following reasons:

(1) In knowledge matching, a simple frequency-based dictionary lookup is used to match knowledge fact from KB. Its accuracy depends on the ambiguity of terms in text and the popularity of matched terms in KB. The higher the matching accuracy, the better the model effect. For this task, we need trade-off between time efficiency and matching precision.

(2) In knowledge transferring, we use a typical transformer [39] to learn the knowledge-enriched sentence representation. Because of the lack of sufficient computational resources, we train the K-XLNet model on datasets for downstream tasks, instead of retraining on large corpus (shown in Figure 3). This may limit its performance to some extent.

(3) Our experiments show that, in domain-specific task, domain knowledge is superior to open-domain knowledge, different knowledge bases have different performances on the same downstream tasks, and the quantity of matched knowledge facts also has an impact on the results (shown in Figure 4). These indicate that injecting relevant explicit knowledge is useful, but the relationship between explicit knowledge and downstream tasks is unknown and worth further study.

5. Conclusion

In this paper, we propose a simple but general knowledge transferring method for language model pretraining. Taking XLNet as a running example, we construct K-XLNet to show

the effectiveness of our method. Extensive experiments show that K-XLNet performs better than XLNet in both open-domain and domain-specific tasks. This work suggests some interesting directions for future work. For example, we can further explore different machine learning techniques to improve the knowledge matching. It would be also interesting to investigate the potential adaptability of explicit knowledge in different downstream tasks.

Data Availability

The databases and test data sets used in this paper are public data, and the data download links can be found in the relevant references.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by General Project of Science and Technology Plan of Beijing Municipal Education Commission (KM202010017011), Program of Beijing Excellent Talents Training for Young Scholar (ZZB2019005), and National Natural Science Foundation of China (42104175).

References

[1] M. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2227–2237, New Orleans, LA, USA, June 2018.

[2] A. Dai and Q. Le, "Semi-supervised sequence learning," in *Proceedings of Neural Information Processing Systems*, pp. 3079–3087, Montreal Canada, December 2015.

[3] S. Singh, E. Hoiem, and D. A. Forsyth, "Swapout: learning an ensemble of deep architectures," in *Proceedings of 2016 Conference on Advances in Neural Information Processing Systems*, pp. 28–36, Barcelona Spain, December 2016.

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[5] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems*, vol. 32, pp. 5754–5764, 2019.

[6] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," 2018, https://arxiv.org/abs/1802.05365.

[7] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does Bert look at? an analysis of Bert's attention," 2019, https://arxiv.org/abs/1906.04341.

[8] F. Petroni, T. Rocktäschel, S. Riedel et al., "Language models as knowledge bases?" in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China, December 2019.

[9] W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov, "Pretrained encyclopedia: weakly supervised knowledge-pretrained language model," 2019, https://arxiv.org/abs/1912.09637.

[10] R. Smith, P. Snow, T. Serry, and L. Hammond, "The role of background knowledge in reading comprehension: a critical review," *Reading Psychology*, vol. 42, no. 3, pp. 214–240, 2021.

[11] C. J. Fillmore, "Frame semantics and the nature of language," in *Proceedings of the Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, pp. 20–32, New York, NY, USA, November 1976.

[12] M. Minsky, *Society of Mind*, Simon & Schuster, New York, NY, USA, 1988.

[13] Y. Yang, J. Cao, J. Shen, R. Yang, and Z. Wen, "Learning analytics based on multilayer behavior fusion," *Blended Learning. Education in a Smart Learning Environment*, Springer, Berlin, Germany, 2020.

[14] K. C. Stevens, "The effect of background knowledge on the reading comprehension of ninth graders," *Journal of Reading Behavior*, vol. 12, no. 2, pp. 151–154, 1980.

[15] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1247–1250, Vancouver Canada, June 2008.

[16] J. Lehmann, R. Isele, M. Jakob et al., "DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[17] S. Elhammadi, L. V. Lakshmanan, R. Ng et al., "A high precision pipeline for financial knowledge graph construction," in *Proceedings Of the 28th International Conference On Computational Linguistics*, pp. 967–977, Barcelona, Spain, September 2020.

[18] P. Colon-Hernandez, C. Havasi, J. Alonso, M. Huggins, and C. Breazeal, "Combining pre-trained language models and structured knowledge," 2021, https://arxiv.org/abs/2101.12294.

[19] Y. Sun, S. Wang, Y. Li et al., "Enhanced representation through knowledge integration," *CoRR*, vol. abs/1904.09223, 2019.

[20] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE.: enhanced language representation with informative entities," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Fortezza da Basso, Italy, July 2019.

[21] W. Liu, P. Zhou, Z. Zhao et al., "K-bert: enabling language representation with knowledge graph," 2019, https://arxiv.org/abs/1909.07606.

[22] C. Rosset, C. Xiong, M. Phan, X. Song, P. Bennett, and S. Tiwary, "Knowledge-aware language model pretraining," 2020, https://arxiv.org/abs/2007.00655.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3111–3119, Lake Tahoe, NV, USA, December 2013.

[24] J. Pennington, R. Socher, and C. D. Manning, *Glove: Global Vectors for Word Representation," Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, 2014.

[25] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "SWAG.: a large-scale adversarial dataset for grounded commonsense inference," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 93–104, Brussels, Belgium, November 2018.

[26] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: a multi-task benchmark and analysis

platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018.

[27] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," 2019, https://arxiv.org/abs/1904.02232.

[28] B. Kim, T. Hong, Y. Ko, and J. Seo, "Multi-task learning for knowledge graph completion with pre-trained language models," in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1737–1743, Barcelona, Spain, December 2020.

[29] P. Lewis, E. Perez, A. Piktus et al., "Retrieval-augmented generation for knowledge-intensive Nlp tasks," 2020, https://arxiv.org/abs/2005.11401.

[30] J. Kaplan, S. McCandlish, T. Henighan et al., "Scaling laws for neural language models," 2020, https://arxiv.org/abs/2001.08361.

[31] G. A. Miller, *WordNet: An Electronic Lexical Database*, MIT press, Cambridge, MA, USA, 1998.

[32] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: an open multilingual graph of general knowledge," in *Proceedings of the AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.

[33] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog,*vol. 1, no. 8, p. 9, 2019.

[34] X. Wang, T. Gao, Z. Zhu et al., "KEPLER.: a unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.

[35] Z. X. Ye, Q. Chen, W. Wang, and Z. H. Ling, "Align, mask and select: a simple method for incorporating commonsense knowledge into language representation models," 2019, https://arxiv.org/abs/1908.06725.

[36] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: issues, techniques, and solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 443–460, 2014.

[37] Z. Fang, Y. Cao, R. Li, Z. Zhang, Y. Liu, and S. Wang, "High quality candidate generation and sequential graph attention network for entity linking," in *Proceedings of the Web Conference 2020, WWW '20*, pp. 640–650, Taipei Taiwan, April 2020.

[38] A. Raganato and J. Tiedemann, "An analysis of encoder representations in transformer-based machine translation," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, November 2018.

[39] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[40] Z. Dong, Q. Dong, and C. Hao, "Hownet and its computation of meaning," in *Proceedings of the COLING 2010, 23rd International Conference on Computational Linguistics, Demonstrations*, pp. 53–56, Beijing, China, August 2010.

[41] B. Xu, Y. Xu, J. Liang et al., "Cn-dbpedia: a never-ending chinese knowledge extraction system," in *Proceedings of the 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE*, vol. 10351, pp. 428–438, Arras, France, June 2017.

[42] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Association for Computational Linguistics, Portland, OR, USA, June 2011.

[43] A. Conneau, R. Rinott, G. Lample et al., "Xnli: evaluating cross-lingual sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2475–2485, Brussels, Belgium, November 2018.

[44] Y. Cao, L. Hou, J. Li et al., "Joint representation learning of cross-lingual words and entities via attentive distant supervision," in *Proceedings of 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, November 2018.

[45] G. A. Levow, "The third international Chinese language processing bakeoff: word segmentation and named entity recognition," in *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing*, pp. 108–117, Association for Computational Linguistics, Sydney, Australia, July 2006.