Knowledge Enhanced Embedding: Improve Model Generalization Through Knowledge Graphs

Anonymous ACL submission

Abstract

Pre-trained language models have achieved excellent results in NLP and NLI, and since the birth of Bert, various new types of Bert have emerged.They are able to grasp the ubiquitous linguistic representational information from large-scale corpora in different ways, but when reading texts, it is difficult for them to combine and use external knowledge to make inferences about other meanings that the text may contain, as people do.To this end, we propose a linguistic model (K2E-BERT) capable of simply incorporating external knowledge, which fuses information from the knowledge graph (triad) with the entity information in the original text.In order to better integrate external knowledge into the original text without letting it deviate from the original meaning of the sentence, we propose a method called EaKA (Entity and Knowledge Align), which can better distance and combine entities and knowledge so that the model can accept new external knowledge without losing the meaning of the original sentence; additionally, we can easily and beyond Bert without changing the internal structure of Bert, we can easily and go beyond the results of BERT, which shows that our approach is feasible.After our experiments, we found good results in several NLP tasks we selected, which indicated that K2E-BERT easily surpassed BERT in generalization ability, proving its effectiveness.

1 Introduction

In recent years, BERT (Devlin et al., 2018) and its variants have achieved many excellent successes in the field of NLP and NLI, where these models can obtain information and representations of human language from a very large open domain corpus in nature. After numerous learning iterations, people are able to analyze entities in a text when they read it, associate them with highly relevant knowledge, and dissect its semantics in context, as shown in Figure 1 . Bert and its varients are pre-trained language model(PLM or PTM).The development of pre-training model (Qiu et al., 2020) can be divided into two stages:pre-train words embedding(PWE) and pre-training context coders(PCE). However, this paper (Sun et al., 2021) summarizes two main shortcomings of the current pre-training model:

(1) The pre-training context encoder has a certain storage capacity;

(2) The knowledge storage of pre-training context encoder has limitations.

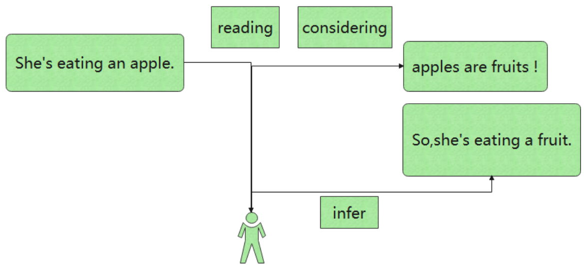

Figure 1: When we read a text, we notice the entities and associate them with knowledge, using inferred knowledge to make more sense of the text.

However, today's pre-trained models can only learn relevant information and representations in the textual ontology, and despite the superior capabilities of these models, this information is limited and it is difficult for the models to uncover the relationships between entities in a large corpus of text. If we can make the models get this human associative ability, then this will allow the models to rise to a new level in generalization ability.

In order to augment the knowledge to the pre-trained language model, the following studies have been done by domestic and foreign scholars respectively:

(1) Adding task-specific knowledge, which can improve the performance of the model on a specific task with high specialization, but not

applicable to other tasks outside the task, such as the GlossBERT (Huang, 2019), which adds the interpretation of certain words to the input of the BERT, only one of which matches the current context, and the output label is whether the word matches that interpretation;

(2) Adding generic knowledge, which maintains the generality of the model but also introduces a part of specific knowledge to the model to improve the performance of the model on some tasks. For example, ERNIE (THU) (Zhang et al., 2019), KnowBERT (Miao, 2019), etc., knowledge is introduced into the model in the pre-training phase.

Knowledge graph, as a semantic network that reveals the relationship between entities, can present the relationship between entities very well. Nowadays, many domain-specific and general domain knowledge graphs have been constructed, e.g., SNOMED-CT (Bodenreider, 2008) used in the medical field, HowNet (Dong et al., 2015) used in Chinese conception. FreeBase (Bollacker, 2008), YAGO (Suchanek et al., 2007) and WordNet (Fellbaum and Miller, 1998) are used in general field. A KG is typically a multi-relational graph containing entities as nodes and relations as edges. Each edge is represented as a triplet (head entity, relation, tail entity) ((h, r, t) for short), indicating the relation between two entities, e.g., (Steve Jobs, founded, Apple Inc.). Despite their effectiveness, how to effectively introduce knowledge into the model is a tricky problem. When introducing external knowledge, the problem of semantic loss is inevitable, and what we want to do is to minimize the loss. So, how do we make good use of the knowledge graph?

Like the problem described in K-BERT (Liu et al., 2019),there are two challenges lies in the road of this knowledge integration:

(1) Heterogeneous Embedding Space (HES): In general, the embedding vectors of words in text and entities in KG are obtained in separate ways, making their vector-space inconsistent;

(2) Knowledge Noise (KN): Too much knowledge incorporation may divert the sentence from its correct meaning. To overcome these challenges, In this paper, we propose a simple transformer bi-directional encoder representation (K2E-BERT) that incorporates external knowledge. K2E-BERT is able to load any pre-trained pre-trained language model such as BERT like K-BERT, because their parameters are the same.

The main contributions of this paper can be summarized as follows.

(1) This paper proposes a method called EaKA to minimize the loss of the original sentence semantics by introducing external knowledge, which enables the model to better incorporate domain knowledge and greatly solves the Heterogeneous Embedding Space (HES) and Knowledge Noise (KN) problem mentioned by K-BERT;

(2) A simpler way of fusing entities with knowledge is used, and a new fused word embedding is added in comparison with the original BERT;

(3) With the subtle injection of KG, K2E-BERT was able to outperform BERT in the only few experiments in the open domain and was able to match and slightly exceed the results of K-BERT in several tasks, and not to change the original structure of Bert.

2 Related Work

Since the introduction of BERT in 2018, many efforts have been made to further optimize it, with most of the research dedicated to the optimization of the process of pre-training with the encoder of BERT.

In terms of optimizing the pre-training process, BERT-WWM (Hu, 2019) uses full word masking instead of single word masking in the corpus to pre-train BERT, and Baidu-ERNIE (Liu, 2019) masks and predicts all entities in the corpus to replace the original pre-training task of BERT. SpanBERT (Levy, 2019) proposes a better Span Masking scheme, and again demonstrates that random masking of consecutive words is better than random masking of scattered words; by adding the Span Boundary Objective (SBO) training target, the performance of BERT is enhanced, especially in some Span-related tasks, such as extractive quizzing. RoBERTa (Stoyanov, 2019), on the other hand, is trained on longer sequences with modified input formats: FULL-SENTENCES + removal of NSP task; changing BERT static masking to dynamic masking; adding a new pre-training dataset CC-NEWS with corpus from 16G text to 160G text;

Text Encoding: using a larger byte-level BPE dictionary.StructBert (Si, 2019)'s main idea is to use language models to find the best arrangement in a series of words and sentences by constructing two new pre-training tasks: Word Structural Objective and Sentence Structural Objective, which disrupt word-level and sentence-level information in the corpus and let the model In this way, the model learns the ability of reconstruction by disrupting the word-level and sentence-level information in the corpus and letting the model predict its original order.

In optimizing the encoder of BERT, XLNet (Zhilin Yang, 2019), a new pre-training goal different from the De-noising Autoencoder approach taken by Bert: Permutation Language Model (PLM for short); this can be understood as how to take specific means to incorporate the bidirectional language model in the autoregressive LM model, and Transformer-XL (Salakhutdinov, 2019) is used to replace the Transformer in BERT to improve its ability to handle long sentences. ERNIE (THU) starts the integration with KG in the pre-training phase, which modifies the encoder of BERT into an aggregator to achieve the mutual integration of words and entities. Specifically, it is stacked by two types of Encoder: T-Encoder and K-Encoder, and the output of T-Encoder and the corresponding knowledge of KG entities are used as the input of K-Encoder. Functionally, the T-Encoder is responsible for capturing lexical and syntactic information from the input sequence; the K-Encoder is responsible for fusing the KG knowledge with the textual information extracted from the T-Encoder, where the KG knowledge is mainly entities here, which are trained by the TransE model.The T-Encoder in THU-ERNIE The structure of T-Encoder in THU-ERNIE is the same as the structure of BERT, and K-Encoder has made some changes. K-Encoder performs Multi-Head Self-Attention operation on the output sequence of T-Encoder and entity input sequence respectively, and then fuses the two through Fusion layer afterwards.These tasks seem to be perfect and have a lot of work, but their improvement is not obvious and consume huge computational resources.

3 How Do We Incorporate External Knowledge into BERT ?

In order to enable the model to incorporate the maximum amount of external knowledge, we propose a process to cope with the problems we face. As shown in Figure 2.

When we get the input sentences, we construct a lookup table for querying between entities and knowledge through the knowledge graph. After obtaining the corresponding entity-knowledge pairs, the entity-knowledge pairs are filled and aligned by our proposed method called EaKA, which solves the problem that embedding knowledge will lose semantics. Further, we use the token ids and knowledge ids obtained by EaKA to reconstruct the input ids and knowledge ids input ids.

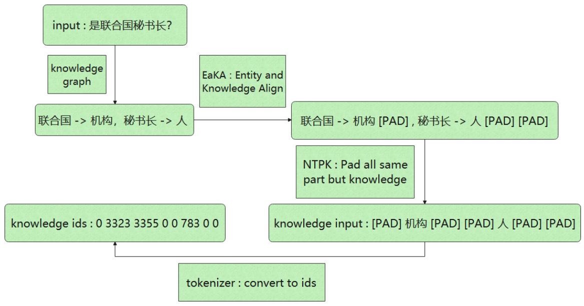

Figure 2: An example showing how we can use knowledge graphs (triples) to extract knowledge and build Knowledge ids.

3.1 EaKA (Entity and Knowledge Align)

In the knowledge graph, the length of entities and knowledge is basically not uniform, which becomes a big stumbling block on the way to combine entities and knowledge. In order to solve this problem, we propose a method named EaKA, which mainly does the following:

(1) Find every possible entity in the sentence and let them match with the entities in the knowledge graph, and if they match, they are the entities we need;

(2) After finding the entities, we can easily get the corresponding knowledge by means of dictionaries. After we have obtained both entities and knowledge, we may find that, for example, a. the length of entities is equal to 4 and the length of knowledge is equal to 5; or b. the length of entities is equal to 4 and the length of knowledge is equal to 2,see Figure 3. These two cases of uneven length are undoubtedly very tricky, and Our proposed strategy is to use the PAD token in the vocab to fill the 'empty space', using this approach to achieve consistency in the length of input ids and knowledge

input ids, and achieve no semantic loss, because the PAD token does not have any semantic information as the filled token. After filling, we get entities and knowledge of the same length! After that, we can iterate through the matched entity-knowledge pairs in the sentence one at a time by a for loop, and then cut-and-merge the original sentence in a circular way to get a new sentence without losing the original meaning step by step and prepare for the next step of building the knowledge sentence.

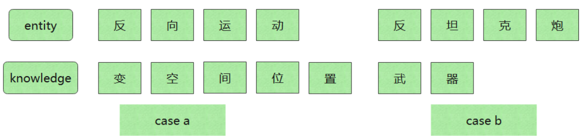

Figure 3: Situations we may encounter when finding entity-knowledge pairs.

3.2 NTPK (Pad all same part but knowledge)

In our later work, we want to add the new input ids to the knowledge input ids by nn.Embedding to get the fused word embedding, so, at the beginning, we thought of taking the knowledge input ids except for the knowledge part, and all the other parts However, this is not feasible, and most importantly, it greatly destroys the distribution of the original input ids by multiplying them by two, which is obviously unreasonable. So, we propose a new approach to solve this problem and prevent the destruction of the original distribution: i.e., still using the PAD token in vocab for filling the same part between input ids and knowledge input ids, again, the PAD token does not have any semantic information! We did the corresponding ablation experiments (only for the optimal parameters): keeping the same fraction vs. not keeping the same fraction, see Table 1. The experimental results show that the effect of using NTPK is better in terms of generalizability of the model.

3.3 Last step but also simple !

We put the resulting input ids and knowledge input ids through the embedding layer to get the implicit vector, and the semantics between them are aligned, so we add them to each other!

As shown in Figure 4, the word embedding and knowledge embedding are added to obtain the fused word embedding incorporating external knowledge, and then added to the remaining two embeddings to obtain the final new bert's embedding.

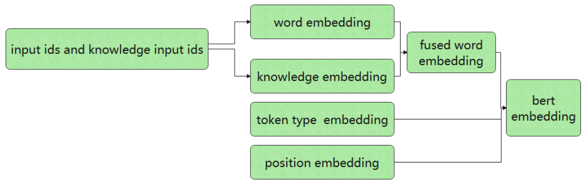

Figure 4: The aligned input ids and knowledge ids are summed to obtain the fused word embedding.

4 Experiments

4.1 Knowledge graph

The knowledge contained in CN-DBpedia is too much due to the lack of equipment resources, so we employ one Chinese KGs, HowNet. (Dong et al., 2015) which is a large-scale language knowledge base for Chinese vocabulary and concepts, in which each Chinese word is annotated with semantic units called sememes. If we take word, contain, sememes as a triple, HowNet is a language KG. Similarly, we refine the official HowNet by eliminating those triples whose entity names are less than 2 in length or contain special characters. The refined HowNet contains a total of 52,576 triples.

4.2 Baselines

In this paper, we compare BERT with K2E-BERT, using the same parameters and the same pre-training weights: Bert-base-Chinese[1] .

4.3 Hyperparameter setting

For a fair comparison of experiments, we use the base-version weights of Bert-base-Chinese, and the parameters set are the same as those in its config. We denote the number of self-attentive layers and heads as L and A, respectively, and the hidden dimension of the embedding vector as H. In detail, we have the following model configuration. L=12, A=12, H=768. The learning rate for all tasks is 2e-5, the decay rate is 0.01, and the warmup ratio is 0.1, epoch is 5, early stop is set to 5, maximum sentence length is 256; where the maximum entity length for different tasks is variable and takes values in the range [1, 18]. The total number of trainable parameters for BERT and K2E-BERT is the same, which means that they are compatible with

[1] https://huggingface.co/bert-base-chinese/tree/main

Models \ Datasets	Book_review		LCQMC		Chnsenticorp		Shopping		Average	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
NTPK	88.11	**87.45**	88.6	**87.41**	**94.92**	95.17	97.01	**96.97**	92.16	**91.75**
Not NTPK	**88.16**	87.04	**89.18**	87.06	94.83	95.17	97.07	96.86	**92.31**	91.53
Max Entity Length	5		7		5		6			

Table 1: Tabel of experimental results comparing the use of NTPK with no use.

Datasets	Train size	Dev size	Test size
LCQMC	238766	8802	12500
Book_review	20000	10000	10000
Chnsenticorp	9600	1200	1200
Shopping	20000	10000	10000

Table 2: Introduction to the size of datasets in the open domain.

each other in terms of model parameters and do not introduce redundant computational overhead.

4.4 Datasets

In this paper, we first compare the performance of KBERT with the BERT on eight Chinese open-domain NLP tasks. Among these four tasks, Book_review, Chnsenticorp, Shopping are single-sentence classification tasks, and LCQMC is the sentence-pair classification tasks:

Book_Review is a online review dataset that contains 20,000 positive and 20,000 negative reviews ;

Chnsenticorp is a hotel review dataset with a total of 12,000 reviews, including 6,000 positive reviews and 6,000 negative reviews;

Shopping is a online shopping review dataset that contains 40,000 reviews, including 21,111 positive reviews and 18,889 negative reviews;

LCQMC is a large-scale Chinese question matching corpus. The goal of this task is to determine if the two questions have a similar intent.

The specific data size of the dataset is shown in Table 2.

4.5 Experimental results

Each of the above datasets is divided into three parts: train, dev, and test. We use the train part to fine-tune the model and then evaluate its performance on the dev and test parts. The experimental results are shown in Table 3.

We can see that the difference between BERT and K2E-BERT on the test dataset is close to one percentage point on the results of the sentence-pair task, LCQMC! And in the other single-sentence tasks also have some improvement on the test dataset, which fully illustrates that K2E-BERT can effectively improve the generalization of the model with the incorporation of external knowledge. Meanwhile, LCQMC has a larger data size compared to the other 3 datasets, which also proves that K2E-BERT brings higher improvement than the case of data scarcity when there is relatively more data, i.e., the size of data is proportional to the effect of generalization improvement it brings to the model.

5 What Kinds of Knowledge Are Beneficial to The Model?

We took the numbers in the interval [4, 9] as input for the parameter max entity length and analyzed the results with the dataset. As shown in Figure 5, the performance results of the test set for each dataset with different max entity lengths are shown.

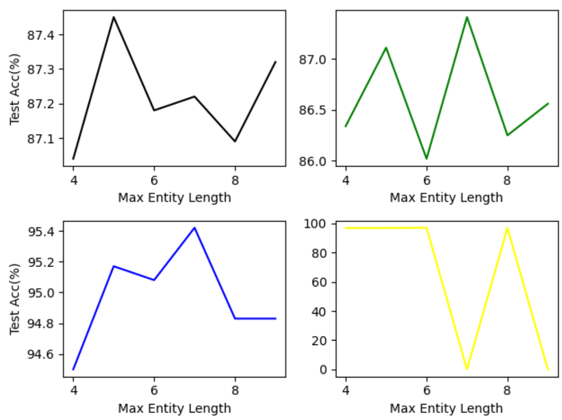

Figure 5: Impact of maximum entity length on the test set. (black is book_review ; green is LCQMC ; blue is chnsenticorp ; yellow is shopping)

As shown in the figure above, the maximum entity length has approximately the same trend on the accuracy of the test set for all datasets.

Models \ Datasets	Book_review		LCQMC		Chnsenticorp		Shopping		Average	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Bert-base-Chinese	**88.56**	87.12	**88.99**	86.17	94.67	95.08	**97.12**	96.84	**92.33**	91.30
K2E-BERT	88.11	**87.45**	88.6	**87.41**	**94.92**	**95.17**	97.01	**96.97**	92.16	**91.75**
Best-entity-length	5		7		5		6			

Table 3: Results of Bert and K2E-Bert on sentence classification tasks on open-domain tasks (Acc. %)

Most of the entities within these intervals have their corresponding knowledge lengths differing from their distances in the range [1, 3], which shows that these entities have the highest improvement in generalization effect when their lengths are not much different from their knowledge lengths.

6 Discussion

So far, we have experimentally demonstrated the effectiveness of K2E-BERT, which can easily incorporate external knowledge and enhance the generalization ability of the original model, allowing the model to learn the function of association, which is a completely new branch of research. For the future work and prospect, we can summarize with the following points:

(1) There are still many low-quality data in the existing knowledge graph. If we can have a higher quality knowledge graph, the generalization ability of K2E-BERT will also be better enhanced;

(2) We will further analyze how other factors in the knowledge graph affect the model and enhance its generalization ability with respect to the relationship of entities in the original text; whether the entities in the original text and their corresponding knowledge in the knowledge graph are somehow related in the semantic space is to be proven later;

(3) Try to transfer the model structure of K2E-BERT to the pre-training task, so that the upstream task can be closer to the downstream task and reduce the performance loss caused by the large gap among them.

7 Conclusion

In this paper, we propose the use of K2E-BERT to implement the fusion of external knowledge into linguistic representations to achieve the ability to associate and reason with the help of knowledge from other domains that people use when reading text. To summarize, K2E-BERT first extracts the entities present in the sentence with the external knowledge map and the knowledge associated with it together, and then uses EaKA to align the entities with the word count of the knowledge so that they have the same word count in space to achieve the effect of reducing the loss of sentence meaning. Next, other tokens that do not exist in the external knowledge graph are replaced with [PAD], aiming to make minimal deviation from the original distribution when obtaining fused word embedding and making changes only in entity positions. Our approach is simpler and useful than K-BERT in facing the challenges of HES and KN. The empirical results show that knowledge graphs are very helpful for NLP and NLI any, and they can improve the generalization ability of the model to a considerable extent. In addition,K2E-BERT incorporates external knowledge and does semantic integration with the original without changing the structure of the BERT model, which allows us to integrate with any existing pre-trained language model and is highly scalable. K2E-BERT is compatible with the model parameters of BERT, which means that users can directly adopt existing pre-trained BERT parameters (e.g., BERT, NeZha (Wei et al., 2019), etc.) on K2E-BERT without the need of pre-training themselves.

References

O. Bodenreider. 2008. Biomedical ontologies in action: role in knowledge management, data integration and decision support. *Yearbook of Medical Informatics*, 17(01):67–79.

K. Bollacker. 2008. Freebase : A collaboratively created graph database for structuring human knowledge. *Proc. SIGMOD' 08*.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Z. Dong, Q. Dong, and I. Ebrary. 2015. Hownet and the computation of meaning. *World Scientific Publishing Co., Inc.*

C. Fellbaum and G. Miller. 1998. *WordNet : an electronic lexical database*. WordNet:An Electronic Lexical Database.

Y CuiW CheT LiuB QinZ YangS WangG Hu. 2019. Pre-training with whole word masking for chinese bert.

L HuangC SunX QiuX Huang. 2019. Glossbert: Bert for word sense disambiguation with gloss knowledge.

M JoshiD ChenY LiuDS WeldL ZettlemoyerO Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans.

W. Liu, P. Zhou, Z. Zhao, Z. Wang, and P. Wang. 2019. K-bert: Enabling language representation with knowledge graph.

Z ZhangX HanZ LiuX JiangM SunQ Liu. 2019. Ernie: Enhanced language representation with informative entities.

P ZhongD WangC Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations.

X. Qiu, T. Sun, Y. Xu, Y. Shao, and X. Huang. 2020. Pre-trained models for natural language processing: A survey. *Science in China: Technical Science in English*, 63(10):26.

Z DaiZ YangY YangJ CarbonellR Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context.

W WangB BiM YanC WuZ BaoL PengL Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding.

Y LiuM OttN GoyalJ DuM JoshiD ChenO LevyM LewisL ZettlemoyerV Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

F. M. Suchanek, G. Kasneci, and G. Weikum. 2007. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *International Conference on World Wide Web*.

Y Sun, Qiu, and Y Zheng. 2021. A survey of knowledge enhancement methods for natural language pretraining models. *Journal of Chinese Information Processing*, 35(7):20.

J. Wei, X. Ren, X. Li, W. Huang, Y. Liao, Y. Wang, J. Lin, X. Jiang, X. Chen, and Q. Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding.

Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yiming Yang Jaime Carbonell Ruslan Salakhutdinov Quoc V. Le Zhilin Yang, Zihang Dai. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.