Knowledge-enabled Question Answering with BERT

Chao Cheng, Annie Feng, Jackson Zhang

Abstract

Knowledge-enabled language representation models are models that are first pre-trained on a large-scale dataset, such as BERT, and then fine-tuned on inputs in the form of knowledge-augmented sentence trees (this is the approach in a paper titled K-BERT). By injecting knowledge from a knowledge graph into a training dataset during the fine-tuning phase, the model is able to achieve better performance on several domain-specific tasks, such as Question Answering. In this paper, we introduce a knowledge-enabled model (BERT + ConceptNet5) for the task of English-language Question Answering, adopting the approach used by K-BERT for Chinese language tasks. In addition, we investigate several optimizations, including knowledge pre-processing and knowledge sampling, which offer increased accuracy and performance over the baseline model.

1 Introduction

Question Answering is the task of providing a suitable answer from within a corpus or set of possible answers when given a natural language query. Recently, large-scale open-domain models that are pre-trained on Wikipedia and other general corpora, such as Google's BERT,[1] have achieved remarkable results on tasks such as Question Answering, Review Classification, and many others. However, while open-domain QA models perform well on general QA, they are unable to understand the highly-specialized and unique language found in the closed-domain question-answering task, where questions are restricted to a specific domain such as law or medicine. For instance, BERT achieves suboptimal results during inference on the electronic medical records (EMR) classification and prediction tasks.[2]

Originating from Tim Berners-Lee's version of a machine-processable web of data, **knowledge graphs** represent rich, flexible, and uniform data that may solve the challenge of closed-domain QA by embedding domain-specific knowledge into traditional language models.[3] In a knowledge graph, knowledge is encoded as a collection of triples in $(object\_1, relation, object\_2)$ format, e.g. `(France, is_a, country)`. By leveraging the ontologies within knowledge graphs, language models such as BERT can learn domain-specific knowledge during the fine-tuning phase, which will lead to a higher level of sophistication across a wide range of tasks within closed-domain areas. To this end, natural language inputs are augmented with knowledge from the knowledge graph in the form of triples to create sentence trees before being fed in during the training phase, thus "injecting" knowledge into the language model.

In 2019, researchers from Peking University and Baidu Research published K-BERT,[2] a knowledge-enabled version of BERT-Chinese that achieved state-of-the-art results on twelve different NLP tasks on Chinese language datasets. In this paper, we employ the same methodologies to train an English version of K-BERT for the question-answering task, including several improvements to account for linguistic differences and variance in the datasets. We also propose some optimizations specific to the question-answering task. After knowledge injection, our model is able to slightly outperform non-knowledge-enabled BERT, indicating that knowledge augmentation is a valid strategy with the potential to achieve even greater results through further investigation and optimization.

2 Motivation and Problem Statement

Pre-training a large language model such as BERT is computationally expensive and inefficient, requiring vast amounts of resources and time. It is also not clear how large scale data from many parts of the web can lead to good models for domain-specific tasks. In this paper, we explore the usage of knowledge graphs as an efficient approach to fine-tuning the model for domain-specific tasks, and as a way to emphasize correlations between domain-specific words to try to get better performance on an English question-answering task.

3 Related Works

In their paper, K-BERT introduced several novel techniques for dealing with problems that commonly arise during the knowledge injection and training process. Primarily, the issue of **knowledge noise** is where too much knowledge embedded into input sentences can obfuscate information and alter the semantic meaning of the original sentence. After knowledge injection, if the sentence is embedded with too many knowledge triples, an abundance of irrelevant knowledge tokens may occur. Although masked language models cannot currently be used as reliable knowledge bases,[4] a mask layer can help control which tokens are visible to each other as a way to encode knowledge in a language model. This method is used by K-BERT, where a *seeing layer* controls token visibility when injecting knowledge from a knowledge graph into input sentences, so that the meaning of the original input sentences is unchanged.

Another issue is heterogenous embedding space, in which inputs to BERT span a particular vector space, whereas knowledge-embedded sentence trees span a different vector space that depends on the implementation (for instance, a tree data structure). To combat this issue, K-BERT transforms inputs during the knowledge injection process into a format matching BERT's embeddings (more details can be found in Section 5.1).

4 Dataset

We train and evaluate our accuracy on the **AI2 Reasoning Challenge 2018**,[5] which has 7,787 genuine grade-school level, multiple-choice science questions. For each question, there are 3 or 4 choices, one of which is correct. In our dataset, we included a question-answer pair and labeled 0 for each wrong answer, and a question-answer pair and label 1 for each correct answer. Also, we evaluate on the "Easy" instead of "Challenge" datasets.

4.1 Comparison to K-BERT

For comparison to K-BERT (Chinese), we used the NLPCC-DBQA dataset for training and evaluation. The NLPCC-DBQA dataset has question-answer pairs labeled 0 or 1, with each question having one best answer (that question-answer pair is labeled 1), and many other suboptimal answers (those question-answer pairs are labeled 0).

5 Knowledge Graph

For our English model, we employed **ConceptNet**, an open, multi-lingual knowledge graph. In this paper, we use a refined subset of ConceptNet[6] as our knowledge graph. We excluded triples including words that are not in the English alphabet or digits 0-9. After this processing, we were left with approximately 3 million out of 34 million triples. We parsed the first 1 million out of these remaining 3 million triples to use. Ideally, we would parse all of the English triples, and create a knowledge graph from these, but practically the parsing was difficult since the triples were encoded in multiple formats from past versions of ConceptNet.

5.1 Comparison to K-BERT

For comparison to K-BERT (Chinese), we used the CNDBpedia knowledge graph in fine-tuning and inference. The authors of K-BERT have selected a refined subset of CNDBpedia for performance on the dataset, and we are using their provided knowledge graph to evaluate our Chinese baseline model.

6 Models and Methodology

We have two baseline models: one for Chinese, and one for English datasets. The Chinese baseline that we will use for comparison is the fine-tuning model implemented by the authors of the K-BERT paper. In their paper, they evaluated their model on classification and named-entity recognition tasks, with several datasets and knowledge graphs. In this paper, we will implement a similar architecture for an English model for the classification task on a question-answering dataset and knowledge graph. Our knowledge graph refinement and dataset selection are similar to the K-BERT paper's choice of the NLPCC-DBQA dataset and CNDBpedia knowledge graph.

6.1 Knowledge injection and fine-tuning

A question-answer pair *sentence* in our dataset is encoded as:

"[CLS] " + *question* + *"[SEP] "* + *answer* + *"[SEP]"*

Our approach to knowledge injection closely follows K-BERT's implementation, summarized as follows. During the knowledge injection process, we augment the original sentence by first converting it into a linked list of tokens, and then

searching each token in a look-up table containing all triples from the knowlege graph. For each token, if any triples containing the token as the primary object are found, then the corresponding $(relation, object\_2)$ branch is appended to the sentence tree at that token. Because certain tokens potentially may have many triples, we limit the maximum number of augmentations per token to two triples. This operation is not performed recursively on any branches, so the maximum depth of any augmented nodes is 2. Additionally, for each augmented token, we mark it as *invisible* to all other tokens in the original sentence besides its parent, and store this information in a $k \times k$ **visible matrix**, where k is the size of the sentence tree. Lastly, after knowledge injection, we are left with the following inputs (including the embeddings used by BERT):

1. knowledge-injected sentence tree
2. visible matrix
3. position embeddings (same as BERT)
4. segment embeddings (same as BERT)

Note that the base BERT model uses only three embedding vectors as input: token embeddings, position embeddings, and segment embeddings. To remedy this, K-BERT implemented a seeing layer which dynamically computes token embeddings based on the visible matrix for each token during the fine-tuning phase.

There are several differences between our knowledge injection and the K-BERT knowledge injection. In our implementation, we modified the above procedure to obtain the correct visible matrix, position, and segment embeddings for English input sentences. Most noticeably, since K-BERT considers tokens to be *characters* in Chinese, whereas English *characters* are letters, we refactored the seeing layer module to instead parse tokens by word. Moreover, during the token segmentation process, we segmented sentences into individual words. However, in the knowledge injection approach employed by K-BERT, knowledge was embedded on *entities* consisting of up to multiple words using the `pkuseg` package.[7] Ideally, we would be able to segment based on semantic meaning, such as "black leaf beetle" being parsed into one token. Instead, we currently segment this phrase into the tokens "black", "leaf", and "beetle", so that each becomes a separate entity rather than a single black leaf beetle entity. Unfortunately, we did not find a suitable English segmentation package that could be easily adapted for this purpose. As a result, the knowledge triples that are injected may not be as relevant, which in turn may affect our results. Additionally, linguistic differences affected our implementation. Chinese nouns are composed of characters, each of which can be standalone words. However, English nouns can be words, which are composed of letters, and are not standalone words. Because of this difference, we must count the token and indices differently, or else we'd be counting letters instead of standalone words.

6.2 Iteration on dataset

We combined the "Easy" and "Challenge" sets for each of *train, dev* and *test* sets. where all incorrect choices are rated 0, and correct choices are rated 1 to turn this into a binary classification task.

We also varied the number of negative and positive examples in *train, dev,* and *test* sets so that our model won't learn to just always predict 0 (which achieves about 75% accuracy since most questions have 3 wrong and 1 correct answer). Our process for this was:

For each question, we include the correct answer. For a question with n incorrect answer choices, with probability $\frac{1}{n}$, we decide to choose i incorrect answers for $i \in \{1...n\}$. Then, out of these n incorrect answer choices, we pick i answer choices without replacement considering all of the n incorrect answer choices equally.

6.3 Iteration on knowledge graph

We excluded triples that included any profanity. Then, we removed triples with irrelevant relations to the AI2 ARC dataset, which contains grade-school science questions. Finally, we parsed the first 1 million out of these remaining 3 million triples to use in our knowledge graph.

These removed relations were "antonym", "capable of", "external url", "distinct from", and "desires". After inspecting the triples that included these relations, we qualitatively reasoned that the triples that contained these relations introduced harmful noise into the model. From the K-BERT paper, this is mentioned as the "knowledge-noise" issue.

For example, we removed these triples that had "desires" as the relation:

	arg1	rel	arg2
420288	person	desires	crispy potato chip
420289	person	desires	critical thinking
420290	person	desires	cry at times
420291	person	desires	create beautiful things
420292	person	desires	control over or destiny

Our hypothesis was that they didn't contribute to correlations we wanted to create for question-answer pairs drawn from grade-school science exams.

7 Results

7.1 Baseline

text_a	text_b	label
黑缘粗角肖叶甲触角有	体长卵形，棕红色；鞘	0
黑缘粗角肖叶甲触角有	头部刻点粗大，分布不	0
黑缘粗角肖叶甲触角有	触角近于体长之半，第	1
黑缘粗角肖叶甲触角有	前胸背板横宽，宽约为	0
黑缘粗角肖叶甲触角有	小盾片舌形，光亮，末	0
黑缘粗角肖叶甲触角有	鞘翅刻点粗大，不规则	0
黑缘粗角肖叶甲触角有	前胸前侧片前缘直；前	0
黑缘粗角肖叶甲触角有	足粗壮；胫节具纵脊，外	0
暮光闪闪的姐姐是谁？	暮光闪闪是一匹雌性独	0
暮光闪闪的姐姐是谁？	她是银甲闪闪(Shinin	1
暮光闪闪的姐姐是谁？	在该系列中，她与最好	0
暮光闪闪的姐姐是谁？	在暮光闪闪成为天角兽	0
暮光闪闪的姐姐是谁？	《我的小马驹：友谊是魔	0
暮光闪闪的姐姐是谁？	动画讲述了一只名叫做	0
暮光闪闪的姐姐是谁？	My Little Pony:Friend	0
暮光闪闪的姐姐是谁？	后成为了天角兽），执行	0
暮光闪闪的姐姐是谁？	她与另外五只小马，共	0
暮光闪闪的姐姐是谁？	每匹小马都分别代表了	0
暮光闪闪的姐姐是谁？	此后，暮光闪闪(Twilig	0

Figure 1: Snippet of NLPCC-DPQA

In their original paper, the authors trained K-BERT on NLPCC-DBQA, using the CNDBpedia knowledge graph. The paper reported an MRR (mean reciprocal rank) score of 94.2 on the test set, and we also independently ran an evaluation of the model on a test dataset, producing the results below (the overall accuracy was 98.16%).

Upon inspection of the NLPCC-DBQA dataset used in the original K-BERT paper, we observed that the number of negative examples vastly outnumbered the number of positive examples. As seen in figure 1, with just 20 examples the ratio of positive to negative examples is $1:10$. In other sections of the dataset, the number of positive examples is even more sparse. This imbalance may explain the high precision, recall, and f1 score on the classification task achieved in the original paper. Because the NLPCC-DBQA dataset used for training and inference is so skewed toward negative examples, the model may simply be learning the counts of the labels rather than the contents of the sentences and thus predict negative labels with higher probability during inference.

Category	Precision	Recall	F1
Correct	0.791	0.859	0.824
Incorrect	0.993	0.988	0.990

Table 1: Results of original K-BERT on Chinese QA

During our initial training and testing, we used the AI2 ARC dataset where each question has one positively labeled example and four negatively labeled examples. Since this dataset this biased towards negative examples, we sought to remedy this issue by creating a more balanced dataset using sampling (see Section 6.2).

7.2 English K-BERT

To evaluate our model, we first measured the baseline performance of BERT without knowledge injection on our dataset. Then, we trained an initial version of K-BERT that utilizes only the knowledge injection module without using a visible matrix. Lastly, we trained a final version of K-BERT, implementing the improvements mentioned in Section 6 as well as including a refactored visible matrix. Each of the models was evaluated on a test dataset consisting of 14,188 samples, and the accuracies are shown below.

Model Configuration	Accuracy
BERT without knowledge	74.94%
K-BERT, no optimization	75.04%
K-BERT, with optimization	75.68%

Table 2: Comparison of Results (English QA)

The figure below shows the decrease in training loss over time.

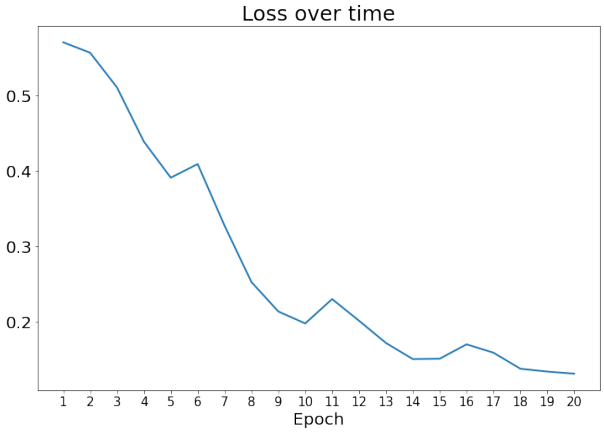

Figure 2: Average Training Loss of English K-BERT

8 Conclusion

In our project, we learned several things:

1. Modeling differences between languages (Chinese vs. English)
2. Effect of data augmentation (by knowledge-injection)
3. Problems with binary classification to prove the effectiveness of a method

For **1**, in the knowledge injection step, we realized that the linguistic and computational difference between Chinese characters vs. English letters could be a modeling issue. The discrete unit of meaning in Chinese is a single character, whereas the smallest unit in English is the word. The Chinese character cannot be broken down further (in handwriting, you can argue strokes, but Chinese character codes for computers don't do this). However, in English, ASCII has the letter as the smallest unit of the language. The representation that our model learns is limited by our initial encoding of the language (ASCII or Chinese character code), so the actual model will have to change when using the Chinese implementation as a starting point, but it's not clear what exactly needs to change. For the project, we did a simple approximation by making each individual Chinese character equivalent to the English word, but this is not always true, as several Chinese characters can map to just one English word.

For **2**, we can characterize our project approach essentially as data augmentation. For question answering, we don't really care about understanding grammatical structure. Instead, the relationships between concepts are more important (the understanding is decoupled from the expression of the understanding). So, a knowledge graph seems like a good choice for data augmentation, since it tries to concisely capture the relationships between concepts by the format of (subject, predicate, object). By augmenting sentences with triples from the knowledge graph, we attempted to emphasize the correlation between related concepts for question-answering. However, further analysis would be required to see if a higher correlation was learned due to knowledge injection. For example, we would like to inspect how the injected knowledge changes the attention compared to the attention over the original sentence.

For **3**, we realized that binary classification was not a good way to evaluate question-answering.

While investigating the KBERT model and implementing modifications for our research goals, we observed an undesirable modeling choice in the original paper, which was the binary classification task as a metric for analyzing model performance. In the paper, the model is performing binary classification to learn "good" responses to questions. As discussed in our results, in replicating the results of the KBERT paper using an English KG and data, we observed that our model accuracy was highly dependent on the ratio of positive and negative labels of question-answer pairs during training and inference. In particular, if the number of negative responses is drastically larger than the number of positive responses, then the model can achieve very high accuracy during inference by simply only outputting negative classification. While inspecting the datasets used in the original paper, we did notice that the number of negatively labeled examples was much larger than that of positively labeled examples, which may have contributed to the high accuracy and precision scores in the paper. In general, due to the high complexity of sentences, a binary classification model is unlikely to truly learn the semantic meaning of sentences other than general correlation. Suggested further work could be doing multi-class classification over answer choices.

8.1 Impact

The main motivation of the research is to explore how we can improve the robustness of current NLP models. Along these lines, we chose to augment the question-answering task with a knowledge graph to try to learn the understanding of concepts through natural language. We also analyzed the pitfalls of

K-BERT, such as their choice of binary classification for evaluating question-answering tasks.

This research explores an alternative to large language models trained on domain-specific corpora without sacrificing the advantages of learning from domain-specific corpora. Existing large pre-trained models use text from the web as training data and lack domain-specific information for question-answering tasks on specific datasets. Moreover, pre-training models for domain-specific tasks require many examples for that model to learn specific relationships. Because knowledge graphs structure information and relationships between pieces of information in a concise format, we can directly enrich information via knowledge injection. As such, our approach avoids costly and indirect domain-specific pre-training and instead fine-tunes inputs by embedding domain-relevant triples tailored to specific datasets to improve performance on the question-answering task.

9 Further Work and Improvements

Instead of classification, we would like to measure K-BERT's performance in the text generation task and analyze perplexity scores in order to better evaluate the model.

In our experiments, we used ConceptNet5 for knowledge injection on the AI2 ARC, a grade school science multiple-choice QA dataset. However, ConceptNet is not a domain-specific knowledge graph for AI2 ARC and may not be as likely to add useful information as would a domain-specific one. For future research, we would like to use parsing APIs like Stanza to create tailored knowledge graphs for specific datasets. For instance, we can create a domain-specific knowledge graph for the AI2 ARC by parsing science textbooks by converting each sentence to an SPO triple.

Lastly, during our token segmentation process, we only segmented sentences into words. In the future, we hope to use the approach employed by the original K-BERT paper and create token entities, consisting of multiple words in order to better capture the semantic meaning of sentences.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2018.

[2] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2901–2908, Apr. 2020.

[3] Andreas Opdahl. Knowledge Graphs and Natural-Language Processing. *Big Data in Emergency Management: Exploitation Techniques for Social and Mobile Data*, pages 75–91, 2020.

[4] Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. Knowledgeable or Educated Guess? Revisiting Language Models as Knowledge Bases. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, August 2021.

[5] https://allenai.org/data/arc.

[6] https://conceptnet.io/.

[7] https://github.com/lancopku/pkuseg-python.