Large-Scale Relation Learning for Question Answering over Knowledge Bases with Pre-trained Language Models

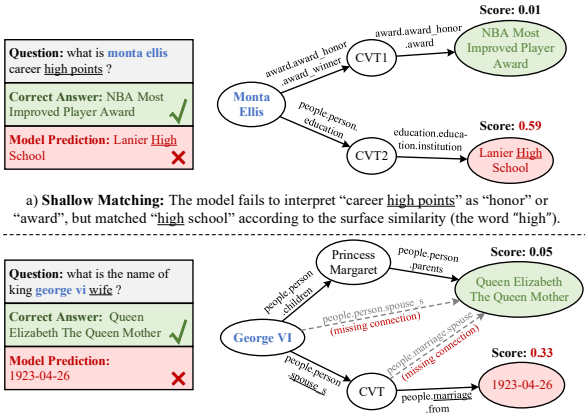
Yuanmeng Yan[1], **Rumei Li**[2], **Sirui Wang**[2], **Hongzhi Zhang**[2], **Daoguang Zan**[3]
Fuzheng Zhang[2], **Wei Wu**[2], **Weiran Xu**[1]*
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Inc., Beijing, China [3]University of Chinese Academy of Sciences, Beijing, China
{yanyuanmeng,xuweiran}@bupt.edu.cn
{lirumei,wangsirui,zhanghongzhi}@meituan.com

Abstract

The key challenge of question answering over knowledge bases (KBQA) is the inconsistency between the natural language questions and the reasoning paths in the knowledge base (KB). Recent graph-based KBQA methods are good at grasping the topological structure of the graph but often ignore the textual information carried by the nodes and edges. Meanwhile, pre-trained language models learn massive open-world knowledge from the large corpus, but it is in the natural language form and not structured. To bridge the gap between the natural language and the structured KB, we propose three relation learning tasks for BERT-based KBQA, including relation extraction, relation matching, and relation reasoning. By relation-augmented training, the model learns to align the natural language expressions to the relations in the KB as well as reason over the missing connections in the KB. Experiments on WebQSP show that our method consistently outperforms other baselines, especially when the KB is incomplete.

1 Introduction

Question Answering over Knowledge Base (KBQA) aims to find the answers to a natural language question given the structured knowledge base (KB) and is widely used in modern question answering and information retrieval systems. Traditional retrieval-based KBQA approaches typically build it as a pipeline system, including name entity recognization, entity linking, subgraph retrieval, and entity scoring. In recent years, with the help of deep representation learning, such approaches have achieved remarkable performance (Dong et al., 2015; Miller et al., 2016; Xu et al., 2016; Sun et al., 2018, 2019; Saxena et al., 2020; He et al., 2021).



a) **Shallow Matching:** The model fails to interpret "career high points" as "honor" or "award", but matched "high school" according to the surface similarity (the word "high").

b) **Incomplete KB:** The model matches "wife" to "spouse" and "marriage". However, some connections are missing from George VI to Queen Elizabeth. On the other hand, the model also fails to reason over the path "George VI – people.person.children – Princess Margaret – people.person.parents – Queen Elizabeth The Queen Mother"

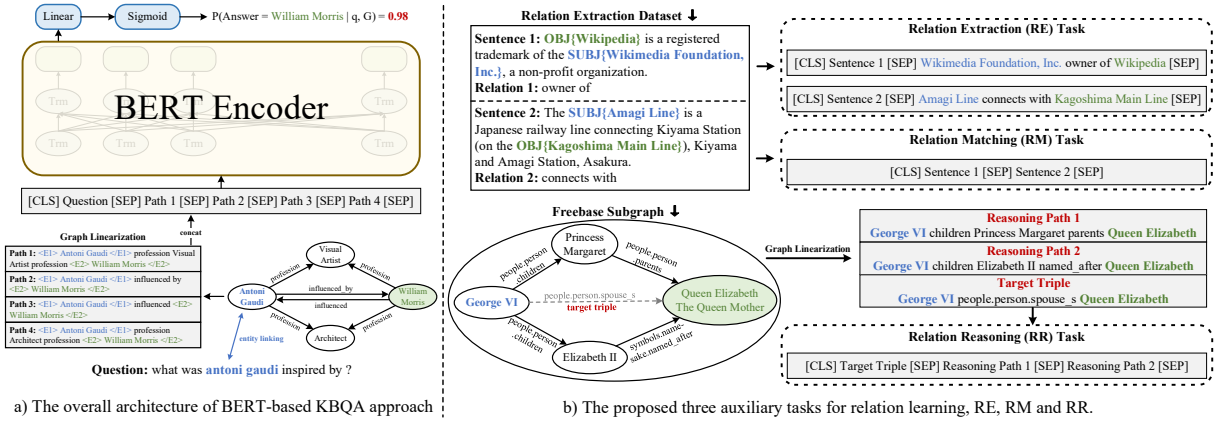
Figure 1: Two error cases from the WebQSP (Yih et al., 2015a) dataset. CVT indicates the Compound Value Type in Freebase. For brevity, we abbreviate the name of some entities and relations.

However, the KBQA task is still challenging especially for multi-hop questions because of two reasons: 1) Due to the complexity of human language, it is often difficult to align the natural language questions with the reasoning paths in the KB. The model tends to learn by surface matching and easily takes shortcut features (Du et al., 2021) for prediction (shown in Figure 1a). 2) In practice, the KB is often incomplete, which also requires the model to reason over the incomplete graph. But the model always fails to do that since it lacks explicit training on reasoning (shown in Figure 1b).

Previous works such as GraftNet (Sun et al., 2018) and PullNet (Sun et al., 2019) mainly solve these problems by introducing external text corpus (e.g. all wikipedia documents) and use specially designed network architecture to incorporate information from the documents. However, the required external resources may be hard to collect in practice. EmbedKGQA (Saxena et al., 2020) solves the KB's incompleteness issue by introducing the pre-trained KB embeddings and trains the ques-

*Work done during internship at Meituan Inc. Weiran Xu is the corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3653–3660
November 7–11, 2021.



a) The overall architecture of BERT-based KBQA approach

b) The proposed three auxiliary tasks for relation learning, RE, RM and RR.

Figure 2: An overview of our approach. For brevity, we abbreviate the name of some entities and relations.

tion embeddings to be fit in the relation embedding space such that they can directly use the scoring function to rank answers. However, their approach mainly grasps the topological structure of the graph but ignores the textual information in entities and relations that should be also useful to score candidate entities.

In this paper, to learn a better mapping from the natural language questions to the reasoning paths in the KB (Gao et al., 2020; Bouraoui et al., 2020), we reformulate the retrieval-based KBQA task to make it a question-context matching form and propose three auxiliary tasks for relation learning, namely relation extraction (RE), relation matching (RM) and relation reasoning (RR). RE and RM both take advantage of the relation extraction datasets, including WebRED (Ormandi et al., 2021) and FewRel (Han et al., 2018). RE trains the model through inferring relations from the sentences, and RM through determining whether two sentences express the same relation. RR constructs the training data from the KB in a self-supervised manner and trains the model to reason over the missing KB connections given the existing paths.

Our contributions can be summarized as follows: 1) To bridge the gap between natural language and the structured KB, we reformulate the KBQA task to be a question-context matching problem and propose auxiliary tasks to enhance the implicit relation learning for pre-trained language models (Devlin et al., 2019). 2) To mitigate the KB's incompleteness issue, we further propose a task for relation reasoning on the KB. 3) Experiments on WebQSP show the effectiveness of our proposed approach, especially when the KB is highly incomplete.[1]

[1]Our code is available at https://github.com/yym6472/KBQARelationLearning

2 Approach

Problem Definition In this paper, we mainly focus on the retrieval-based KBQA. Given an input query q, we first annotate the named entities in the query and link them to the nodes in the KB. Then some heuristic algorithm[2] is applied to retrieve a query-specified subgraph $\mathcal{G} = \{\langle e, r, e' \rangle | e, e' \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ is the set of all candidate entities that probably contains the answer of q, and $\mathcal{R}$ denotes the relation set. Our task is to calculate a score s_i for each candidate entity $e_i \in \mathcal{E}$ indicating whether e_i is the answer entity or not.

In this section, we first present how to solve KBQA with BERT, then we introduce three proposed auxiliary tasks to augment the relation learning for BERT.

2.1 BERT for KBQA

For each question q, we can obtain its topic entity e_{topic} from the entity linking system.[3] Then, as shown in Figure 2a, we convert the candidate entity scoring problem into a question-context matching task as follows.

We first find all paths in $\mathcal{G}$ that connect the topic entity e_{topic} and the candidate entity e_i. We set a maximum number of paths[4] and apply down-sampling when the number exceeds the threshold. Then we construct the textual form of each path by replacing the nodes with entity names and the edges with relation names in the KB. Finally, we concatenate the question q and all paths $p_1, ..., p_n$ to make an input sample x_i =

[2]Following previous works (Sun et al., 2018), we use the Personalized PageRank algorithm (Haveliwala, 2002).

[3]It is guaranteed that $e_{\text{topic}} \in \mathcal{G}$ when retrieving subgraphs. For samples without linked topic entity, we remove them from the train set and count them as wrong cases when testing.

[4]The number is set to 10 in our experiments.

[CLS] q [SEP] p_1 [SEP] ... p_n [SEP].

Here, we regard these paths as the facts between the topic entity e_{topic} and the candidate entity e_i. We aim to use BERT to predict whether the hypothesis "e_i is the answer of q" is supported by those KB facts. Thus, we feed the sample to BERT and take the representation corresponding to [CLS] token for binary classification:

$$s_i = \sigma(\boldsymbol{w}^\top \texttt{BERT}_{\texttt{CLS}}(x_i)) \quad (1)$$

$$\mathcal{L}_i = -(y \cdot \log s_i + (1-y) \cdot \log(1-s_i)) \quad (2)$$

, where σ is the sigmoid function and y is ground truth label indicating whether e_i is the answer entity of q or not.

2.2 Auxiliary Tasks for Relation Learning

The performance of KBQA depends heavily on the mapping from the natural language questions to the relations in the path. To further enhance the relation learning of BERT, we propose three auxiliary tasks for relation learning, as shown in Figure 2b.

Relation Extraction (RE) One straightforward idea is to use the relation extraction dataset, where the model learns to extract the relation expressed in the sentence between the given head and tail entity. Similar to KBQA, we concatenate the sentence and the one-hop path to construct an RE example for BERT: [CLS] s [SEP] h, r, t [SEP], where s, h, r and t indicates sentence, head entity, relation and tail entity respectively.

Moreover, to simulate the 2-hop reasoning in KBQA, we also combine two RE examples to make a compositional one: [CLS] s_1, s_2 [SEP] $h_1, r_1, t_1(h_2), r_2, t_2$ [SEP], where the tail entity of the first example is same to the head entity of the second example.

Relation Matching (RM) In relation matching task, we assume that two sentences with the same relation should have similar representations. Thus, we concatenate two sentences and train BERT through predicting whether two sentences express the same relation: [CLS] s_1 [SEP] s_2 [SEP], where the label is 1 if s_1 and s_2 express the same relation and 0 otherwise.

Relation Reasoning (RR) BERTRL (Zha et al., 2021) proposes a self-supervised approach for KB completion task. They choose one triplet (h, r, t) from the KB and assume it is missing. Then they find other multi-hop paths from h to t, and use them to predict whether (h, r, t) exists in the KB or not: [CLS] h, r, t [SEP] p_1 [SEP] ... p_n [SEP] By training on BERTRL, the model learns to reason and complete the missing connections, which is extremely helpful for KBQA on the incomplete KB.

Training Since all three auxiliary tasks are formulated as a binary classification task and only differ in the data construction phase, we can either use them to pre-train BERT before KBQA (noted as *pre-train*) or train them jointly with KBQA in a multi-task paradigm (noded as *joint*). In our experiments, we find both settings work well and produce similar results (see Section 3.4 for more details).

3 Experiments

3.1 Datasets

KBQA Dataset To evaluate the effectiveness of our approach, we conduct experiments on WebQuestionsSP (WebQSP, Yih et al. 2015a) dataset. It contains 4737 questions that are answerable using Freebase. Following Sun et al. (2018), we reserve 250 examples from the training set for tuning hyperparameters and early stopping, resulting in 2848/250/1639 examples for training, validation, and test respectively.

We obtain and preprocess WebQSP using the scripts[5] released by Sun et al. (2018). It mainly includes entity linking and subgraph retrieval in two steps. The entity linking results are directly taken from the codebase[6] released by Yih et al. (2015b). For each question, there is a set of seed entities[7] and will be used in the subgraph retrieval phase. The subgraphs are retrieved through the Personalized PageRank (PPR) algorithm (Haveliwala, 2002), and we set the max number of entities in each subgraph to 500. Among the 1639 examples in the test set, the answers of 120 questions are not retrieved from the subgraph, so the answer coverage is about 92.68% in the subgraph retrieval phase.

Relation Extraction Datasets In the relation learning tasks, we use WebRED (Ormandi et al., 2021) and FewRel (Han et al., 2018) dataset as external resources. For more details about these datasets and how we process them to construct relation learning tasks, please refer to Appendix A.

[5] https://github.com/OceanskySun/GraftNet/tree/master/preprocessing

[6] https://github.com/scottyih/STAGG

[7] It can be an empty set if the named entity recognization or entity linking fails.

3.2 Baselines

We compare our approach to several baselines, including KV-Mem (Miller et al., 2016), GraftNet (Sun et al., 2018), PullNet (Sun et al., 2019), EmbedKGQA (Saxena et al., 2020) and NSM (He et al., 2021). Please refer to Appendix B for more details. Besides, we also provide results of BERT (without additional relation learning) as a baseline to show the effectiveness of our proposed relation learning tasks.

3.3 Metrics

When evaluating our model, we first feed each linearized input to BERT and get the corresponding score between 0 to 1. For each question, we rank all candidate entities in the subgraph by the scores and calculate the hits@1 and F1 as follows:

- **Hits@1** If the highest-ranked entity is the answer entity, then hits@1 is 1. Otherwise, hits@1 is 0.
- **F1 score** Given a threshold, we consider all candidate entities whose scores are greater than the threshold as the answers predicted by the model. Then we calculate the F1 score between the ground truth answer entities and the model predicted answer entities. In our experiments, we select the threshold that performs best in the validation set.

Then we average the Hits@1 and F1 scores over all test examples. For questions whose answers are not covered by the retrieved subgraph, we regard them as wrong predictions. Note that we treat hits@1 as the primary metrics, since the results of F1 score show a large variance due to its sensitivity to the threshold. We provide more training details in Appendix C.

3.4 Main Results

The experimental results are shown in Table 1. We find that the results with BERT outperform most of the baselines (except for the NSM). When comparing to PullNet, BERT achieves a relative improvement of 4.6%, demonstrating the effectiveness of solving KBQA with BERT.

On the other hand, the results with all three relation learning tasks (72.3) significantly outperform the BERT baseline (71.2), showing that the proposed auxiliary tasks benefit the relation matching and relation reasoning of BERT.

Model	dev set	test set	
	Hits@1	F1	Hits@1
Baselines			
KV-Mem†	-	38.6	46.7
GraftNet‡	-	62.4	66.7
PullNet†	-	-	68.1
EmbedKGQA†	-	-	66.6
NSM†	-	67.4	**74.3**
Our implementation			
BERT	71.0	63.4	71.2
$\text{BERT}_{\text{+RE}}$ *pre-train*	68.1	62.1	72.8*
$\text{BERT}_{\text{+RM}}$ *pre-train*	71.8	63.4	72.6*
$\text{BERT}_{\text{+RR}}$ *pre-train*	69.8	61.8	71.7*
$\text{BERT}_{\text{+RE,RM,RR}}$ *pre-train*	69.4	62.5	72.3*
$\text{BERT}_{\text{+RE}}$ *joint*	67.3	57.4	72.4*
$\text{BERT}_{\text{+RM}}$ *joint*	72.2	64.5	**72.9***
$\text{BERT}_{\text{+RR}}$ *joint*	67.3	62.9	71.2
$\text{BERT}_{\text{+RE,RM,RR}}$ *joint*	71.8	60.0	72.0*

Table 1: Experimental results on WebQSP dataset. Baseline results with † are taken from He et al. (2021), while results with ‡ are taken from Sun et al. (2018). The numbers with * indicate the significant improvement over the BERT baseline with $p < 0.05$ under t-test.

Ablation Studies To check which task contributes to the final result most, we conduct experiments where only one task is applied at a time. From the second part of Table 1, we can observe that RE and RM are the two most contributing tasks, and even training with them individually can outperform training with all three tasks together. Meanwhile, RR also brings performance improvement (from 71.2 to 71.7) under the *pre-train* setting, but its improvement is not as significant as RE and RM. This may be because the model doesn't require much reasoning ability under the full KB setting.

Pre-training or Joint Training When comparing the *pre-training* setting with *joint training*, we find both settings work well and outperform the BERT baseline. For RE and RR, *pre-training* seems better than *joint training*, while for RM, *joint training* is slightly better.

3.5 Analysis

Results over the Incomplete KB To verify the robustness of our approach when the KB is incomplete, we randomly remove 50% of the KB facts in the retrieved subgraphs and conduct experiments on this incomplete version of the WebQSP dataset.

The results[8] are illustrated in Figure 3. We can

[8] Appendix D provides more results under the incomplete KB (with different proportions) as well as the comparison to baselines.

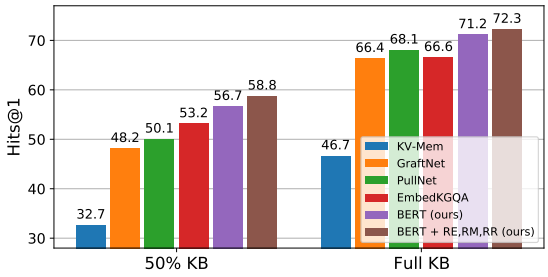


Figure 3: The performance comparison with the full KB and the 50% KB. We only compare to baselines that also report results with 50% KB.

Model	Annotation Type		
	none	*all*	*head-tail*
BERT	69.8	70.9	71.2
BERT$_{+RE,RM,RR}$ *joint*	70.1	71.3	72.0

Table 2: Hits@1 results on WebQSP with different annotation types of entity spans.

observe that: 1) Our approach consistently outperforms other baselines under both the full KB and the 50% KB settings. 2) With 50% KB, adding relation learning tasks achieve more performance gain than with full KB (+2.1 vs +1.1), demonstrating that our relation learning tasks are especially useful when the KB is incomplete.

Annotation of Entity Spans As discussed in Soares et al. (2019), different markers for entity spans have a great impact on the BERT-based relation extraction task. To find out the best annotation strategy for KBQA, we conduct experiments with three types of annotations: 1) Using no annotation (noted as *none*). 2) Using `<E>` and `</E>` to annotate all entities in the reasoning paths (noted as *all*). 3) Using `<E1>` and `</E1>` to annotate all head entities and using `<E2>` and `</E2>` to annotate all tail entities (noted as *head-tail*).

As shown in Table 2, we find *none* performs worst while *head-tail* achieves the best result. We can conclude that the annotations of the entity spans are still required for the BERT model. They bring structural information that helps the model to identify the entity. Meanwhile, fine-grained annotations (*head-tail*) are better than the coarse-grained ones (*all*).

Influence of Negative Samples In our experiments, we want to speed up the training by downsampling negative samples of KBQA. However, as shown in Table 3, we find that the performance is also related to the number of negative samples. In general, more negative samples will bring a higher Hits@1 score. One potential solution to this issue is hard negative mining, and we will leave it for future work.

# Neg. Samples	20	50	100	200	500
F1 score	60.6	62.3	63.5	63.2	63.8
Hits@1	65.0	69.0	69.7	70.7	72.0

Table 3: Results on WebQSP when downsampling negative samples during training.

4 Conclusion

In this paper, we propose three auxiliary tasks to augment relation learning for BERT-based KBQA method, including relation extraction, matching and reasoning. These tasks not only bridge the gap between the natural language and the structured KB, but also explicitly train the model to reason over the incomplete KB. The experimental results on WebQSP demonstrate the effectiveness of our approach, especially when the KB is incomplete.

Acknowledgements

We thank all anonymous reviewers for their helpful comments and suggestions. This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

Broader Impact

KBQA is a widely applied technique in natural language processing, especially in question answering and information retrieval tasks. This work focuses on the retrieval-based approaches and proposes to use BERT-like pre-trained language models to improve the scoring function for ranking the candidates. Though our approach takes advantage of the learned open-world knowledge in BERT and achieves better results, the introduction of pre-trained language models may lead to some potential risks such as introducing extra data biases and being sensitive to adversarial examples. On the other hand, our proposed relation extraction and relation matching tasks use external resources (i.e. the relation extraction datasets) that may contain the ethical risk. Therefore, users should pay special attention when preparing these resources and guarantee they are task-relevant, unbiased, and ethical.

References

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 260–269.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of nlu models. *arXiv preprint arXiv:2103.06922*.

Tianyu Gao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2020. Neural snowball for few-shot relation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7772–7779.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 553–561, New York, NY, USA. Association for Computing Machinery.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409.

Robert Ormandi, Mohammad Saleh, Erin Winter, and Vinay Rao. 2021. Webred: Effective pretraining and finetuning for relation extraction on the web. *arXiv preprint arXiv:2102.09681*.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and 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)*, pages 2380–2390.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015a. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015b. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.

Hanwen Zha, Zhiyu Chen, and Xifeng Yan. 2021. Inductive relation prediction by bert. *arXiv preprint arXiv:2103.07102*.

Model	10% KB		30% KB		50% KB		Full KB	
	F1	Hits@1	F1	Hits@1	F1	Hits@1	F1	Hits@1
Baselines								
KV-Mem	4.3	12.5	13.8	25.8	21.3	33.3	38.6	46.7
GraftNet	6.5	15.5	20.4	34.9	34.3	47.7	62.4	66.7
PullNet	-	-	-	-	-	50.3	-	68.1
EmbedKGQA	-	-	-	-	-	53.2	-	66.6
NSM	-	-	-	-	-	-	**67.4**	**74.3**
Our implementation								
BERT	12.95	25.89	30.88	44.84	41.81	56.72	63.39	71.18
$\text{BERT}_{+\text{RE}}$	13.51	26.79	31.09	46.51	42.19	58.28	62.05	72.77
$\text{BERT}_{+\text{RM}}$	**13.72**	26.79	**31.23**	46.57	**43.09**	**59.02**	**64.51**	**72.89**
$\text{BERT}_{+\text{RR}}$	13.65	**26.98**	29.59	**46.63**	42.10	57.11	61.83	71.73
$\text{BERT}_{+\text{RE,RM,RR}}$	13.58	26.79	30.50	46.57	41.42	58.77	62.46	72.28

Table 4: Experimental results on WebQSP dataset. The baseline results are taken from their corresponding paper. 10%, 30% and 50% indicate the incomplete KB settings, where the facts in the subgraphs are randomly removed to 10%, 30% and 50%. For our implemented results, we report the best results among *pre-train* and *joint* settings.

A Data Processing for Relation Learning

Dataset Details WebRED[9] is a relation extraction dataset based on WikiData. It is firstly constructed through distant supervision and then denoised by human annotators. WebRED contains more than 500 relations in WikiData and releases 107819/3898 denoised examples for train/test. We further remove those contradictory examples (i.e. the number of positive raters is equal to the number of negative raters) and obtain 107761/3898 examples for train/test.

FewRel[10] includes 80 relations in WikiData and 700 examples for each relation, resulting in totally 56,000 examples. Among 80 relations, 64/16 relations are split for training/test.

Data Processing For Relation Extraction (RE) task, we directly use the negative examples in the WebRED dataset for negative sampling. For Relation Matching (RM) task, we randomly sample one sentence with the same relation label to construct the positive pair and randomly sample 9 sentences with other relation labels to construct negative pairs. For Relation Reasoning (RR) task, we use all retrieved subgraphs from the WebQSP's *train* split and run the script[11] released by Zha et al. (2021) to generate training samples.

[9]Available at https://github.com/google-research-datasets/WebRED

[10]Available at http://www.zhuhao.me/fewrel/index.html

[11]https://github.com/zhw12/BERTRL

B Baselines

We compare our approach to the following baselines:

KV-Mem (Miller et al., 2016) adopts a key-value memory network to store the KB facts and uses it to augment the open domain question answering.

GraftNet (Sun et al., 2018) propose to solve open domain question answering task by retrieving from the KB and the textual corpus and design a variant of graph convolution network for the heterogeneous graph.

PullNet (Sun et al., 2019) uses GraftNet as the model architecture but it also learns how to retrieve information and expand the subgraph during the training and test phase.

EmbedKGQA (Saxena et al., 2020) uses the pre-trained KB embeddings and trains the question encoder to make question embeddings aligned with the relation embedding space such that they can directly use the scoring function to predict whether a given entity is the answer or not.

NSM (He et al., 2021) propose to use the neural state machine (NSM) to solve the KBQA task and uses bidirectional hybrid reasoning and a two-stage teacher-student architecture to augment the reasoning ability of the student model.

C Training Details

We run all our experiments on one single NVIDIA Tesla V100 (32GB) GPU. We set the batch size to 128 and set the max sequence length to 128

for BERT. We evaluate the model every 1000 or 3000 training steps depending on the number of total training steps in one epoch, and the evaluation takes about 6 minutes. We train the model for up to 3 epochs and use a learning rate of 2e-5. For the pre-trained BERT, we download the `bert-base-uncased` model from HuggingFace[12], and set the dropout rate to 0.2 during training. The best results are typically achieved after training BERT for 2-3 epochs (roughly 15,000 - 25,000 steps), which often takes 6-8 hours (roughly 1.8 steps per second) for training. The number of model parameters is 109,483,009 (109M), including the parameters of BERT and the linear head for binary classification. For all hyperparameters used in our experiments, we manually tune them on the reserved 250 train examples of WebQSP. F1-score is used as the metric to select the best hyperparameters.

D More Results over the Incomplete KB

We show more experimental results on incomplete KB in Table 4. We can make the following observations: 1) When the KB is extremely incomplete (10% KB and 30% KB), our approach can achieve significant performance gain compared to previous work GraftNet (Sun et al., 2018) (+7.2 on 10% KB setting and +11.5 on 30% KB setting). 2) The relation reasoning (RR) task is well performed when the KB is extremely incomplete (10% KB and 30% KB), but the performance gain decreases when the KB is relatively complete (50% KB and Full KB). 3) The relation matching (RM) task is the most robust task that shows very strong performance gain with different KB's incompleteness.

[12] https://huggingface.co/bert-base-uncased