Retrieval, Re-ranking and Multi-task Learning for Knowledge-Base Question Answering

Zhiguo Wang, Patrick Ng, Ramesh Nallapati, Bing Xiang
AWS AI Labs
{zhiguow, patricng, rnallapa, bxiang}@amazon.com

Abstract

Question answering over knowledge bases (KBQA) usually involves three sub-tasks, namely topic entity detection, entity linking and relation detection. Due to the large number of entities and relations inside knowledge bases (KB), previous work usually utilized sophisticated rules to narrow down the search space and managed only a subset of KBs in memory. In this work, we leverage a *retrieve-and-rerank* framework to access KBs via traditional information retrieval (IR) method, and re-rank retrieved candidates with more powerful neural networks such as the pre-trained BERT model. Considering the fact that directly assigning a different BERT model for each sub-task may incur prohibitive costs, we propose to share a BERT encoder across all three sub-tasks and define task-specific layers on top of the shared layer. The unified model is then trained under a multi-task learning framework. Experiments show that: (1) Our IR-based retrieval method is able to collect high-quality candidates efficiently, thus enables our method adapt to large-scale KBs easily; (2) the BERT model improves the accuracy across all three sub-tasks; and (3) benefiting from multi-task learning, the unified model obtains further improvements with only 1/3 of the original parameters. Our final model achieves competitive results on the SimpleQuestions dataset and superior performance on the FreebaseQA dataset.

1 Introduction

Answering natural language questions by searching over large-scale knowledge bases (KBQA) is highly demanded by real-life applications, such as Google Assistant, Siri, and Alexa. Owing to the availability of large-scale KBs, significant advancements have been made over the years. One main research direction views KBQA as a *semantic matching* task (Bordes et al., 2014; Dong et al., 2015; Dai et al., 2016; Hao et al., 2017; Mohammed et al., 2018; Yu et al., 2018; Wu et al., 2019; Chen et al., 2019a; Petrochuk and Zettlemoyer, 2018), and finds a relation-chain within KBs that is most similar to the question in a common semantic space, where the relation-chain can be 1-hop, 2-hop or multi-hop (Chen et al., 2019b). Another research direction formulates KBQA as a *semantic parsing* task (Berant et al., 2013; Bao et al., 2016; Luo et al., 2018), and tackles questions that involve complex reasoning, such as ordinal (e.g. What is the second largest fulfillment center of Amazon?), and aggregation (e.g. How many fulfillment centers does Amazon have?). Most recently, some studies proposed to derive answers from both KBs and free-text corpus to deal with the low-coverage issue of KBs (Xu et al., 2016; Sun et al., 2018; Xiong et al., 2019; Sun et al., 2019). In this paper, we follow the first research direction since the relation-chain type of questions counts the vast majority of real-life questions (Berant et al., 2013; Bordes et al., 2015; Jiang et al., 2019).

Previous semantic matching methods for KBQA usually decompose the task into sequential sub-tasks consisting of topic entity detection, entity linking, and relation detection. For example in Figure 1, given the question "Who wrote the book Beau Geste?", a KBQA system first identifies the topic entity "Beau Geste" from the question, then the topic entity is linked to an entity node (m.04wxy8) from a list of candidate nodes, and finally the relation *book.written_work.author* is selected as the relation-chain leading to the final answer. Previous methods usually worked on a subset of KB in order to fit KB into memory. For entity linking, some sophisticated heuristics were commonly used to collect entity candidates. For relation detection, previous work usually enumerated all possible 1-hop and 2-hop relation-chains (starting from linked entity nodes) as candidates. All

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 347–357
April 19 - 23, 2021.

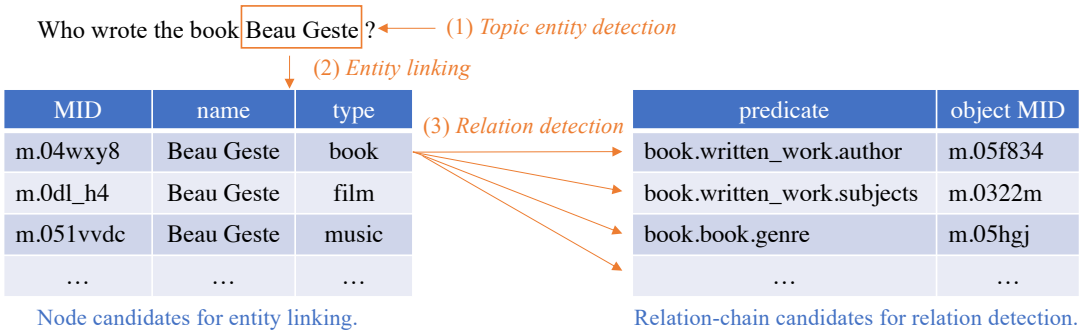

Figure 1: A typical workflow for KBQA. Given a question "Who wrote the book Beau Geste?", the topic entity detection model first identifies a topic entity "Beau Geste" from the question. Then, the entity linking model links the topic entity into an entity node (m.04wxy8) in the KB. Finally, the relation *book.written_work.author* is selected as the relation-chain leading to the final answer node (m.05f834).

these workarounds may prevent their methods from generalizing well to other datasets, and scaling up to bigger KBs.

To tackle these issues, we leverage a *retrieve-and-rerank* strategy to access KBs. In the *retrieval* step, we ingest KBs into two inverted indices: one that stores all entity nodes for entity linking, and the other one that stores all subject-predicate-object triples for relation detection. Then, we use TF-IDF algorithm to retrieve candidates for both entity linking and relation detection sub-tasks. This method naturally overcomes the memory overhead when dealing with large-scale KBs, therefore makes our method easily scale up to large-scale tasks. In the *re-ranking* step, we leverage the advanced BERT model to re-rank all candidates by fine-grained semantic matching. For the topic entity detection sub-task, we utilize another BERT model to predict the start and end positions of a topic entity within a question. Since assigning a different BERT model for each sub-task may incur prohibitive costs, we therefore propose to share a BERT encoder across sub-tasks and define task-specific layers for each individual sub-task on top of the shared layer. This unified BERT model is then trained under the multi-task learning framework. Experiments on two standard benchmarks show that: (1) Our IR-based retrieval method is able to collect high-quality candidates efficiently; (2) the BERT model improves the accuracy across all three sub-tasks; and (3) benefiting from multi-task learning, the unified model obtains further improvements with only 1/3 of the original parameters. Our final model achieves competitive results on the SimpleQuestions dataset and superior performance on the FreebaseQA dataset.

2 Task Definition

Knowledge-base question answering (KBQA) aims to find answers for natural language questions from structural knowledge bases (KB). We assume a KB $\mathcal{K}$ is a collection of subject-predicate-object triples $\langle e_1, p, e_2 \rangle$, where $e_1, e_2 \in \mathcal{E}$ are entities, and $p \in \mathcal{P}$ is a relation type between two entities, $\mathcal{E}$ is the set of all entities, and $\mathcal{P}$ is the set of all relation types. Given a question Q, the goal of KBQA is to find an entity node $a \in \mathcal{E}$ from the KB as the final answer, thus can be formulated as

$$\hat{a} = \arg\max_{a \in \mathcal{E}} Pr(a|Q, \mathcal{K}) \quad (1)$$

where $Pr(a|Q, \mathcal{K})$ is the probability of a to be the answer for Q. A general purpose KB usually contains millions of entities in $\mathcal{E}$ and billions of relations in $\mathcal{K}$ (Bollacker et al., 2008), therefore directly modeling $Pr(a|Q, \mathcal{K})$ is challenging. Previous studies usually factorize this model in different ways. One line of research forms KBQA as a *semantic parsing* task $Pr(q|Q, \mathcal{K})$ to parse a question Q directly into a logical form query q, and execute the query q over KB to derive the final answer. Another line of studies views KBQA as a *semantic matching* task, and finds a relation-chain within KB that is similar to the question in a common semantic space. Then the trailing entity of the relation-chain is taken as the final answer. Following this direction, we decompose the KBQA task into three stages: (1) identify a topic entity t from the question Q, where t is a sub-string of Q; (2) link the topic entity t to a topic node $e \in \mathcal{E}$ in KB; and (3) detect a relation-chain $r \in \mathcal{K}$ starting from the topic node e, where r can be 1-hop, 2-hop or multi-hop relation-chains within KB. Correspond-

ingly, we factorize the model as

$$\begin{aligned} Pr(a|Q,\mathcal{K}) &= Pr(t,e,r|Q,\mathcal{K}) \\ &= P_t(t|Q,\mathcal{K})P_l(e|t,Q,\mathcal{K}) \\ &\quad P_r(r|e,t,Q,\mathcal{K}) \quad (2) \end{aligned}$$

where $P_t(t|Q,\mathcal{K})$ is the model for topic entity detection, $P_l(e|t,Q,\mathcal{K})$ models the entity linking process, and $P_r(r|e,t,Q,\mathcal{K})$ is the component for relation detection stage. We will discuss how to parameterize these components in Section 4.

3 Background

We briefly introduce some background required by the following sections.

BERT: BERT model (Devlin et al., 2019) follows the multi-head self-attention architecture (Vaswani et al., 2017), and is pre-trained with a masked language modeling objective on a large-scale text corpus. It has achieved state-of-the-art performance on a bunch of textual tasks. Specifically, for semantic matching tasks, BERT simply concatenates two textual sequences together, and encodes the new sequence with multiple self-attention layers. Then, the output vector of the first token is fed into a linear layer to compute the similarity score between the two input textual sequences.

Freebase: We take Freebase (Bollacker et al., 2008) as our back-end KB to answer questions. It contains more than 46 million topic entities and 2.6 billion triples. Each entity has an internal machine identifier (MID) and a set of aliases. Some entities also have properties such as entity types and detailed descriptions. Freebase contains a special entity category called *Compound Value Type* (CVT), which does not have a name or alias, and is only used to collect multiple fields of an event or a special relationship. In the official Freebase dump [1], all facts are formulated as the unified subject-predicate-object triples, and there is no explicit split for entities and relations. We partition facts in Freebase into a set of entities $\mathcal{E}$ and a set of relations $\mathcal{K}$ by following the pre-processing steps in Chah (2017).

Inverted Index and TF-IDF: *Inverted index* is an optimized data structure of finding documents (from a large document collection) where a query word X occurs. It is commonly used for fast free-text searches. Term Frequency-Inverse Document Frequency (TF-IDF) is a ranking function usually used together with an inverted index to estimate the relevance of documents to a given search query (Schütze et al., 2008).

[1] https://developers.google.com/freebase

4 Retrieval and Re-ranking for KBQA

In this section, we describe how to parameterize P_t, P_l and P_r in Equation (2).

4.1 Topic Entity Detection Model P_t

The goal of a topic entity detection model $P_t(t|Q,\mathcal{K})$ is to identify a topic entity t that the question Q is asking about, where t is usually a substring of Q. Previous approaches for this task can be categorized into two types: (1) rule-based and (2) sequence labeling. The rule-based approaches take all entity names and their alias from a KB as a gazetteer, and n-grams of the question that exactly match with an entry in the gazetteer are taken as topic entities (Yih et al., 2015; Yao, 2015; He and Golub, 2016; Yu et al., 2017). The advantage of this method is that no machine learning models need to be involved. However, the drawbacks include: (1) topic entities need to have the exact same surface strings as they occur in KB, and (2) memory-efficient data structures need to be designed to load the massive gazetteer into memory (Yao, 2015). Other approaches leverage a sequence labeling model to tag consecutive tokens in the question Q as topic entities (Dai et al., 2016; Bordes et al., 2015; Mohammed et al., 2018; Wu et al., 2019). This approach is able to predict more precise topic entities, thus prunes some unimportant matched entities.

Inspired from the Start/End prediction method commonly utilized for machine reading comprehension tasks (Wang and Jiang, 2016; Seo et al., 2016), we cast the topic entity detection task into predicting the start and end positions of the topic entity t in the question Q. Formally, we denote t_s and t_e as the start and end positions for a topic entity t, and assume this process is independent of KB. Thus the model can be further decomposed as $P_t(t|Q,\mathcal{K}) = P_s(t_s|Q)P_e(t_e|Q)$, where $P_s(t_s|Q)$ and $P_e(t_e|Q)$ are the probabilities of t_s and t_e to be the start and end positions. This formulation directly models the goal of the topic entity detection task, i.e. finding the best topic entity within a question, therefore can give a more precise estimation.

We leverage the advanced BERT model to parameterize $P_s(t_s|Q)$ and $P_e(t_e|Q)$. Concretely, we

first leverage BERT encoder to encode the input question Q, then apply two independent linear layers (with one output neuron) on top of BERT's output for each token. The start/end scores are normalized across all tokens with the $softmax$ function to estimate the probabilities of each token position to be the start/end of the topic entity.

4.2 Entity Linking Model P_l

The purpose of an entity linking model $P_l(e|t, Q, \mathcal{K})$ is to link the recognized topic entity t to an entity node $e \in \mathcal{E}$ in KB. A general purpose KB usually contains millions of nodes in $\mathcal{E}$, which makes it almost impossible to search over the full space. Previous methods usually narrow down the search space based on some heuristic rules. For example, Yih et al. (2015) and Wu et al. (2019) used keyword search to collect all nodes that have one alias exactly matching the topic entity, and Yin et al. (2016) collected all nodes that have at least one word overlapping with the topic entity. Once a smaller set of candidates is selected, complicated neural networks can be utilized to compute the similarity between a candidate node and the topic entity in the question context.

Inspired from the recent success of question answering over free-text corpus (Chen et al., 2017; Wang et al., 2018, 2019), we propose a *retrieve-and-rerank* method to solve the entity linking task in two steps. In the first *retrieval* step, we create an inverted index for all entity nodes, where each node is represented with all tokens from its aliases and description. Then, we use the topic entity t as a query to retrieve top-K candidate nodes from the index with the TF-IDF algorithm[2]. The similar method is also used by Vakulenko et al. (2019) and Nedelchev et al. (2020). This information retrieval (IR) method is better than previous work in the following ways. First, our method can find candidate nodes even if a topic entity does not have an exactly matched entity node. Second, we do not have to maintain all entity nodes inside CPU memory, and can still query candidates efficiently, which enables our method to be easily adapted to large-scale KBs. Third, the relative importance of various matched words is naturally considered in the TF-IDF algorithm.

In the second *re-ranking* step, we leverage BERT model to compute the similarity between each candidate node and the topic entity in the given question context. Concretely, we represent each pair of a topic entity t and a candidate node e as a sequence of tokens with the format " `[CLS] topic entity [SEP] question pattern [SEP] node name [SEP] node types [SEP] node description [SEP]`", where `topic entity` is the string for the topic entity t, `question pattern` is the question string with t being removed, `node name`, `node types` and `node description` are the name, types and description for the topic node e, and `[SEP]` is the delimiter used by BERT model. We encode this sequence with BERT model, then feed the hidden vector for the token `[CLS]` into a linear layer (with one output neuron) to compute the similarity score for each pair of t and e.

4.3 Relation Detection Model P_r

The relation detection model $P_r(r|e, t, Q, \mathcal{K})$ traverses relation-chains starting from a linked topic node e, and attempts to detect the correct relation-chain r that answers the question Q. Previous work usually enumerates all possible 1-hop and 2-hop relation-chains starting from a linked topic node e, and leverages deep neural networks to compute semantic similarity between each candidate relation-chain r and the question Q (Bordes et al., 2014; Yih et al., 2015; Dong et al., 2015; Yu et al., 2017; Wu et al., 2019). In real KBQA systems, usually, a list of linked nodes from the entity linking step is considered to retain high recall. If we enumerate all relation-chains for all these linked topic nodes, we will end up with a large collection of candidate relation-chains. Furthermore, re-ranking so many candidate relation-chains will add much run-time latency, especially when a heavy model such as BERT is utilized.

To address this issue, we propose to use the *retrieve-and-rerank* method for the relation detection task, and deal with this task in two stages similar to what we did for the entity linking task. In the first *retrieval* step, we create an inverted index for all subject-predict-object triples, where each triple is represented as all tokens from the name of the subject entity, the name of the predicate, and types of the object entity. Then, we use the question Q as a query to retrieve top-K 1-hop relation-chains with the constraint that all subject nodes are from the list

[2] Mohammed et al. (2018) also created an inverted index for all nodes. However, they generated ngrams of each entity name into several entries, and looked up exactly matched ngram candidates by keyword searching.

of linked entity nodes. If two-hop relation-chains are required in a target dataset, we will do the same querying step again, but with the constraint list being all object entities from the first retrieval step. We acknowledge that this method does not consider the already covered semantics in the first retrieval step, when we do the second step retrieval. Since the main goal of the retrieval step is to collect a list of high-quality candidates, we will perform better semantic matching in the re-ranking step with more powerful neural networks. If multi-hop relation-chains are needed, we can iterate this process until reaching the maximum steps. Usually, the number of max-hop is pre-computed on the target question sets. Another way is to utilize a model to decide when to stop (Chen et al., 2019b), however we will leave this option in the future work.

After collecting a list of relation-chains, we leverage another BERT model to compute the similarity between a question Q and each relation-chain r. Each pair of Q and r will be represented as a sequence of tokens with the format "`[CLS] question [SEP] topic-entity name [SEP] relation chain [SEP] answer name [SEP] answer types [SEP]`", where `topic-entity name` is the name for the linked entity node, `relation chain` is the word sequence of a candidate relation-chain[3], `answer name` is the name of the trailing node in the relation-chain, and `answer types` are all types of the trailing node. The hidden vector for the `[CLS]` token will be fed into a linear layer (with one output neuron) to predict the similarity between Q and r.

5 Multi-Task Learning for KBQA

5.1 Training Objectives

For the topic entity detection model, we define the objective function as the cross-entropy loss between true distributions and predicted distributions. We sum up the cross-entropy losses for both start and end models, and average over all N training instances:

$$L(\theta_t) = -\frac{1}{N}\sum_{i=1}^{N} \log(P_s^i) + \log(P_e^i) \quad (3)$$

where θ_t is the trainable parameter for topic entity detection model.

[3]A relation-chain is split into a word sequence based on delimiters such as periods, hyphens and underscores.

Both entity linking and relation detection tasks are ranking tasks, therefore we leverage a hinge loss function for both tasks:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \max(0, l + s(Q, c^-) - s(Q, c^+) \quad (4)$$

Where θ is the trainable parameter, l is a margin, $s(Q, c)$ can be the model of P_l or P_r, c^+ is a correct candidate, and c^- is an incorrect candidate. We set $l = 1.0$ in this work.

5.2 Multi-Task Learning

A naive approach would be to use three different BERT encoders for the topic entity detection, entity linking and relation detection sub-tasks individually. Since BERT model is a very large model, it is expensive to host three BERT models in real applications. To address this, we propose to share a BERT encoder across all three sub-tasks, and define lean layers for each individual sub-task on top of the shared layer. This unified model is then trained under the multi-task learning framework proposed by Liu et al. (2019). First, training instances for each sub-tasks are packed into mini-batches separately. At the beginning of each training epoch, mini-batches from all three sub-tasks are mixed together and randomly shuffled. During training, a mini-batch is selected, and the model is updated according to the task-specific objective for the selected mini-batch.

6 Experiments

We evaluate the effectiveness of our model on standard benchmarks in this section. We first conduct experiments on each sub-task with a separate BERT model in Section 6.2, 6.3 and 6.4, then evaluate the influence of sharing a BERT encoder for all three models in Section 6.5. Finally, we benchmark our method on full Freebase in Section 6.6.

6.1 Datasets and Basic Settings

We evaluate our proposed model on two large-scale benchmarks: SimpleQuestions and FreebaseQA. Other existing datasets, such as WebQuestions (Berant et al., 2013), Free917 (Cai and Yates, 2013) and WebQSP (Yih et al., 2016), are not considered, because they only contain few thousands of questions which is even less than the number of relation types in Freebase.

SimpleQuestions: The SimpleQuestions dataset (Bordes et al., 2015) is so far the largest

KBQA dataset. It consists of 108,442 English questions written by human annotators, and all questions can be answered by 1-hop relation chains in Freebase. Each question is annotated with a gold-standard subject-relation-object triple from Freebase. We follow the official train/dev/test split. To fairly compare with previous work, we leverage the released FB2M subset of Freebase as the back-end KB for this dataset. FB2M includes 2M entities and 5k relation types between these entities.

FreebaseQA: FreebaseQA dataset (Jiang et al., 2019) is a large-scale dataset with 28K unique open-domain factoid questions which are collected from triviaQA dataset (Joshi et al., 2017) and other trivia websites. Each question can be answered by a 1-hop or 2-hop relation-chain from Freebase. All questions have been matched to subject-predicate-object triples in Freebase, and verified by human annotators. Comparing with other KBQA datasets, FreebaseQA provides more linguistically sophisticated questions, because all questions are created independently from Freebase. FreebaseQA also released a new subset of Freebase, which includes 16M unique entities, and 182M triples. We follow the official train/dev/test split, and take the Freebase subset as the back-end KB for this dataset.

Basic Settings: We leverage the pre-trained BERT-base model with default hyper-parameters in our experiments. We create inverted indices for topic nodes and relations with Elasticsearch[4], and utilize the BM25 (a variance of TF-IDF) algorithm to retrieve inverted indices.

6.2 Topic Entity Detection Experiments

In order to train and evaluate our topic entity detection model, we annotate the ground-truth topic entity for each question with the following steps. First, for each question, all alias names for the annotated topic entity MID are collected from Freebase. Then, we match each alias against the question string. If more than one alias occurs in the question string, the longest matched string will be annotated as the ground-truth. Otherwise, the span with the minimum edit distance will be selected as the ground-truth.

We implement a BERT-based sequence labeling model as a baseline for our Start/End prediction model described in Section 4.1. The baseline model follows the same architecture for the named entity recognition (NER) task in Devlin et al. (2019), where we use BIO schema to annotate each question token. Since the sequence labeling method may predict multiple spans to be topic entities, we choose the span with the maximum average token score as the final prediction.

Models	SimpleQ. EM	SimpleQ. F_1	FreebaseQA EM	FreebaseQA F_1
BIO	94.9	97.3	65.1	75.2
Start/End P_t	**96.4**	**97.8**	**74.3**	**81.5**
Multi-task P_t	96.0	97.7	70.6	79.3

Table 1: Results for topic entity detection.

We employ the metrics exact match (EM) and F_1 proposed in Rajpurkar et al. (2016) to evaluate the identified topic entities. Experimental results are shown in Table 1. We can see that our Start/End prediction model works better than the BIO sequence labeling baseline. Specifically, in FreebaseQA dataset, since the questions are longer and more complicated, our Start/End model outperforms the BIO sequence labeling model by a large margin.

6.3 Entity Linking Experiments

We retrieve a list of candidate nodes for each question as follows. For questions in the training sets, we use the ground-truth topic entity as the query to retrieve top-100 candidate nodes. For questions in the dev and testing sets, we use top-N predicted topic entities as queries, and retrieve top-50 candidates for each topic entity. All candidates are then sorted based on their popularity (number of out-going triples). Based on the results on dev sets, we set N=1 for the SimpleQuestions dataset, and N=5 for the FreebaseQA dataset. We employ the top-K accuracy to evaluate entity linking results, where an instance is correct if there is at least one correct candidate inside the top-K candidate list.

Retrieval step: We implement a *Keyword-search* baseline for the retrieval step. In this baseline, all nodes, having an alias exactly matching with the topic entity, are collected as candidates. All candidates are sorted based on their popularity, i.e. the number of out-going triples. Table 2 lists the results of the baseline as well as our IR-based method proposed in Section 4.2. Our IR-based method gets better results than the Keyword baseline on both datasets. The main reason is that our

[4]https://www.elastic.co/products/elasticsearch

	SimpleQuestions		FreebaseQA	
	Keyword	IR	Keyword	IR
Top-1	24.4	75.7	39.0	35.3
Top-5	55.0	86.7	70.1	72.7
Top-10	68.8	89.2	76.1	81.1
Top-50	89.3	**93.7**	81.8	89.4
Top-100	92.7	93.7	82.9	**89.8**

Table 2: Qualitative analysis on entity linking candidates for the retrieval step.

IR-based method does not require exact matches between predicted topic entities and entity nodes within KB, therefore is more robust to prediction errors or entity name variances from the up-streaming topic entity detection model.

Re-ranking step: We feed top-100 candidate nodes from the retrieval step into our entity linking model P_l to re-rank all candidates. Table 3 shows results on the SimpleQuestions dataset. The first group of numbers in Table 3 are results from previous state-of-the-art models. We can see that our entity linking model P_l outperforms previous models in terms of Top-1 accuracy, and achieves competitive results in terms of Top-10 and Top-20 accuracy. Table 4 lists the results of our model and previous work on the FreebaseQA dataset. Our entity linking model P_l improves accuracy over previous work (Wu et al., 2019) by a large margin. Since top-5 predicted topic entities are used for the FreebaseQA dataset, we create another ranker to multiply together scores from both the topic entity detection model and entity linking model, and list the results in the row P_tP_l in Table 4[5]. The P_tP_l ranker gets even better Top-1 accuracy than our entity linking model P_l alone, which verifies that our factorization in Equation (2) is reasonable.

6.4 Relation Detection Experiments

We retrieve a list of relation-chain candidates for each question as follows. For questions in the training sets, we use the correct entity node as the start point to search top-100 candidates. For questions in the dev and testing sets, we use the top-N entity nodes predicted by our entity linking model as start points to retrieve top-100 candidates. For candidates with the same subject and relation type, we sort them based on the popularity of the trailing object node (number of in-coming triples), and only keep top-4 relation-chains in the final list. Based on the results on the dev set, we set N=30 for the SimpleQuetions dataset and N=10 for the FreebaseQA dataset. For the SimpleQuestions dataset, since all questions can be answered with 1-hop relation-chains, we only retrieve 1-hop candidates. For the FreebaseQA dataset, following the method in Jiang et al. (2019), we only expand 1-hop relation-chain candidates into 2-hop candidates if the object node of a 1-hop relation-chain is a CVT node. For the SimpleQuetions dataset, a prediction is correct if both the subject and relation are correctly retrieved. For the FreebaseQA dataset, a prediction is correct if the final answer matches with the ground-truth answer.

Retrieval step: We implement a baseline to collect all relation-chains starting from entity nodes, and sort all relation-chains based on their popularity, i.e. the in-coming triples for the trailing object. Retrieval results from the baseline are listed in the "All" columns in Table 5. The results from our IR based method (proposed in Section 4.3) are shown in the "IR" columns in Table 5. The last row "Rel/Q" in Table 5 gives the average number

Models	Top-1	Top-10	Top-20
Yin et al. (2016)	72.7	86.9	88.4
Yu et al. (2017)	79.0	89.5	90.9
Qiu et al. (2018)	81.1	91.7	93.4
Wu et al. (2019)	82.2	**92.5**	**93.6**
P_l	84.2	92.1	93.1
Multi-task P_l	**84.3**	92.1	93.1
Full Freebase	79.0	88.9	90.3

Table 3: Entity linking results on SimpleQuestions.

Models	Top-1	Top-3	Top-5
Wu et al. (2019)	52.4	79.6	85.7
P_l	69.4	**84.8**	**86.6**
P_tP_l	**71.9**	84.6	86.3
Multi-task P_l	68.1	84.2	85.8
Multi-task P_tP_l	71.7	84.7	86.4
Full Freebase	68.1	81.6	83.8

Table 4: Entity linking results on FreebaseQA.

[5]Accuracy of the P_tP_l model is not given in Table 3, because only the best (top-1) topic entity is used for retrieving entity candidates in the SimpleQuestions dataset.

	SimpleQuestions		FreebaseQA	
	All	IR	All	IR
Top-1	16.5	52.8	0.3	10.9
Top-5	53.5	80.8	1.4	20.3
Top-10	65.6	86.1	3.4	26.6
Top-50	81.9	91.7	22.8	49.8
Top-100	87.6	**92.5**	31.9	**62.6**
Rel/Q	772	100	3021	100

Table 5: Qualitative analysis for relation-chain candidates in the retrieval step, where "Rel/Q" is the average number of relation-chains per question.

of relation-chains per question. Comparing the "IR" columns with "All" columns, our IR-based method retrieves fewer relation-chains but maintains better recall.

Re-ranking step: We feed top-100 relation-chain candidates from the retrieval step into our relation detection model P_r to re-rank all candidates. Table 6 shows the results from previous state-of-the-art models as well as our relation detection model P_r. We can see that our P_r model obtains very competitive results on the SimpleQuestions dataset, and outperforms previous models by a large margin in the FreebaseQA dataset. We also create a model $P_tP_lP_r$ to multiply scores from our topic entity detection model, entity linking model and relation detection model. By considering the influence of all three components, our $P_tP_lP_r$ model achieves even better accuracy on the FreebaseQA dataset.

6.5 Multi-task Learning Experiments

Our method achieves very strong performance by leveraging three BERT encoders for each model component. In this section, we share a BERT encoder for all three models, and jointly train the unified model with the multi-task learning method described in Section 5.2. Experimental results from this model are shown in rows with the prefix "Multi-task" in Table 1, 3, 4, and 6. Although the multi-task model only has about 1/3 of the original parameters, it is able to achieve better end-to-end accuracy in Table 6, and retain similar performance as before on the other two sub-tasks.

6.6 KBQA over Full Freebase

Most of the previous studies conducted KBQA experiments with a subset of Freebase, because it is

Models	SimpleQ.	FreebaseQA
Dai et al. (2016)	75.7	N/A
Yin et al. (2016)	76.4	N/A
Yu et al. (2017)	77.0	N/A
Wu et al. (2019)	77.3	37.0
Hao et al. (2018)	**80.2**	N/A
Petrochuk (2018)	78.1	N/A
P_r	79.4	45.4
$P_tP_lP_r$	79.4	49.1
Multi-task P_r	79.7	47.9
Multi-task $P_tP_lP_r$	79.7	**51.7**
Full Freebase	74.1	35.4

Table 6: Relation detection accuracy in the end-to-end manner.

hard to fit the full Freebase into memory (Bordes et al., 2014; Dong et al., 2015). Our method ingests Freebase into inverted indices on hard disk storage, thus naturally overcomes the memory overhead. This advantage enables us to evaluate our method on the full Freebase. The last rows of Table 3, 4, and 6 show the results of running our "Multi-task" model over the full Freebase. Significant degradations are observed in entity linking and relation detection tasks on both datasets. This phenomenon reveals that previous studies may overestimate the capacity of their KBQA models. We suggest that researchers evaluate their models on the full Freebase in the future.

7 Conclusion

In this work, we proposed a *retrieve-and-rerank* strategy to access large-scale KBs in two steps. First, we leveraged traditional IR methods to collect high-quality candidates from KBs for entity linking and relation detection. Second, we utilized the advanced BERT model to re-rank candidates by fine-grained semantic matching. We also employed a BERT model to predict the start and end positions of the topic entity in a question. To reduce the model size, we proposed a joint model to share BERT encoder across all three sub-tasks, and create task-specific layers on the top. We then trained this joint model with multi-task learning. Experimental results show that our method achieves superior results on standard benchmarks, and is able to scale up to large-scale KBs.

References

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514, Osaka, Japan. The COLING 2016 Organizing Committee.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620, Doha, Qatar. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 423–433, Sofia, Bulgaria. Association for Computational Linguistics.

Niel Chah. 2017. Freebase-triples: A methodology for processing the freebase data dumps. *arXiv preprint arXiv:1712.08707*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2019a. Bidirectional attentive memory networks for question answering over knowledge bases. 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 2913–2923, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019b. UHop: An unrestricted-hop relation extraction framework for knowledge-based question answering. 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 345–356, Minneapolis, Minnesota. Association for Computational Linguistics.

Zihang Dai, Lei Li, and Wei Xu. 2016. CFO: Conditional focused neural question answering with large-scale knowledge bases. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 800–810, Berlin, Germany. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Beijing, China. Association for Computational Linguistics.

Yanchao Hao, Hao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Pattern-revising enhanced simple question answering over knowledge bases. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3272–3282.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231, Vancouver, Canada. Association for Computational Linguistics.

Xiaodong He and David Golub. 2016. Character-level question answering with attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1598–1607, Austin, Texas. Association for Computational Linguistics.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. Freebaseqa: A new factoid qa data set matching trivia-style question-answer pairs with freebase. In *Pro-*

ceedings 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 318–323.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint arXiv:1904.09482*.

Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Zhu. 2018. Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2185–2194, Brussels, Belgium. Association for Computational Linguistics.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana. Association for Computational Linguistics.

Rostislav Nedelchev, Debanjan Chaudhuri, Jens Lehmann, and Asja Fischer. 2020. End-to-end entity linking and disambiguation leveraging word and knowledge graph embeddings. *arXiv preprint arXiv:2002.11143*.

Michael Petrochuk and Luke Zettlemoyer. 2018. SimpleQuestions nearly solved: A new upperbound and baseline approach. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Yunqi Qiu, Manling Li, Yuanzhuo Wang, Yantao Jia, and Xiaolong Jin. 2018. Hierarchical type constrained topic entity detection for knowledge base question answering. In *Companion Proceedings of the The Web Conference 2018*, pages 35–36. International World Wide Web Conferences Steering Committee.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*, page 260.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

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, Hong Kong, China. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Svitlana Vakulenko, Javier David Fernandez Garcia, Axel Polleres, Maarten de Rijke, and Michael Cochez. 2019. Message passing for complex question answering over knowledge graphs. In *CIKM*, pages 1431–1440. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R 3: Reinforced ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. 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 5877–5881, Hong Kong, China. Association for Computational Linguistics.

Dekun Wu, Nana Nosirova, Hui Jiang, and Mingbin Xu. 2019. A general fofe-net framework for simple and effective question answering over knowledge bases. *arXiv preprint arXiv:1903.12356*.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete KBs with knowledge-aware reader. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4258–4264, Florence, Italy.

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, Berlin, Germany. Association for Computational Linguistics.

Xuchen Yao. 2015. Lean question answering over Freebase from scratch. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 66–70, Denver, Colorado. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. 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.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple question answering by attentive convolutional neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1746–1756, Osaka, Japan.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581, Vancouver, Canada. Association for Computational Linguistics.

Yang Yu, Kazi Saidul Hasan, Mo Yu, Wei Zhang, and Zhiguo Wang. 2018. Knowledge base relation detection via multi-view matching. In *European Conference on Advances in Databases and Information Systems*, pages 286–294. Springer.