Pattern-revising Enhanced Simple Question Answering over Knowledge Bases

Yanchao Hao[1,2], **Hao Liu**[1], **Shizhu He**[1], **Kang Liu**[1,2], **Jun Zhao**[1,2]
[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
{yanchao.hao, hao.liu2017, shizhu.he, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Question Answering over Knowledge Bases (KB-QA), which automatically answer natural language questions based on the facts contained by a knowledge base, is one of the most important natural language processing (NLP) tasks. Simple questions constitute a large part of questions queried on the web, still being a challenge to QA systems. In this work, we propose to conduct pattern extraction and entity linking first, and put forward pattern revising procedure to mitigate the error propagation problem. In order to learn to rank candidate subject-predicate pairs to enable the relevant facts retrieval given a question, we propose to do joint fact selection enhanced by relation detection. Multi-level encodings and multi-dimension information are leveraged to strengthen the whole procedure. The experimental results demonstrate that our approach sets a new record in this task, outperforming the current state-of-the-art by an absolute large margin.

1 Introduction

As the amount of the knowledge bases (KBs) grows, such as DBpedia[1], Freebase[2], and WikiData[3], people are paying more attention to seeking effective methods for accessing these precious intellectual resources. While knowledge bases are usually very large and not easily accessible for users. KB-QA (Unger et al., 2014), which takes natural language as query language, is a more user-friendly solution, and has become a research focus in recent years. At the same time the growing amount of data has led to a heterogeneous data landscape where QA systems struggle to keep up with the volume, variety and veracity of the underlying knowledge.

Simple question answering over knowledge bases, which answers a question using a single fact in knowledge bases, is not simple at all and far from being solved. This is the setup of the SimpleQuestions benchmark recently presented by Bordes et al. (2015). The task of KB-QA for simple questions could be put as follows. Let $\mathcal{G} = (s_i, p_i, o_i)$ be a background knowledge base represented as a set of triples, where s_i represents a subject entity, p_i a predicate (also denoted as relation), and o_i an object entity. Given a natural language question represented as an utterance $q = w_1, ..., w_n$, the task of simple question answering is to find a triple $(\hat{s}, \hat{p}, \hat{o}) \in \mathcal{G}$ such that $\hat{o}$ is the intended answer for question q. This task can be formulated to finding the best matches of subject $\hat{s}$ and predicate $\hat{p}$.

$$\hat{s}, \hat{p} = \arg\max_{s,p \in \mathcal{G}} p(s, p|q) \tag{1}$$

The setup of using a single fact to answer a question seems simple, while there are many existing challenges faced by QA systems. One challenge is the extremely complex language used in question utterance. On one hand, in many different ways natural language is used to express the same information need. On the other hand, the same word used in different sentences may express different meaning. For example, two questions "*who created Apple Inc.*" and "*who started Apple*" have the same meaning and

[1] http://dbpedia.org

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

[3] https://www.wikidata.org

Proceedings of the 27th International Conference on Computational Linguistics, pages 3272–3282
Santa Fe, New Mexico, USA, August 20-26, 2018.

in order to retrieve the correct answer we need to map the meaning conveyed by them to the "*founder*" relation in KB. The word "*Apple*" in the questions refers to the "*Apple Corporation*" instead of the fruit "*apple*". Natural language is complex while KB is much more universal. Another challenge is the vast amount of facts in large scale KB.

KB is quite large to some extent. KB has a huge number of fact triples and entities. A common and effective way to remain only a small subset of facts and entities is to conduct entity linking of a question over KB firstly. There are some work conducting entity linking by searching n-gram words of a question among all entity names (Bordes et al., 2015; Golub and He, 2016). While some other work propose a special-purpose sequence labeling network to focus on more probable candidates in question utterance, then linking them to entities (Dai et al., 2016; Yin et al., 2016b). Previous approaches assume that their preceding steps are correct to produce a good result for next procedure. While it should be pointed out that both the labeling and the entity linking process may lead to error propagation problems. For example, if the previous steps provide a wrong labeling result or don't recall the gold subject entity in the generated candidates, the following procedure can't make a good choice to retrieve the gold subject and predicate pair.

Faced with these problems and following previous work, we propose to do pattern extraction and entity linking first to extract possible entity mention and question pattern[4], and reduce the large number of candidate entities in KB. Due to the performance of the pattern extraction step, we observe that there is a certain proportion of bad cases in the extracted mention-pattern pairs. So we propose a novel pattern revising procedure to improve the quality of the extracted patterns. Then the extracted mentions are used to identify candidate entities in KB that the question refers to. This is the entity linking procedure. The candidate fact pool is formed by incorporating all the predicates connected to the corresponding subjects. The next step is to select the gold subject and predicate from the candidate fact pool.

For one question, there may be a large set of candidate entities. It is unrealistic to use all of them to generate candidate facts in practice. In this case, we need to let the candidate entities which are more likely to generate the gold fact get ahead of the rest. Previous methods rarely considered the information contained in the question pattern or candidate entities when doing subject selection. So we conduct relation detection, which identifies the KB predicate that a question utterance refers to, to reorder the entity linking results. Enhanced entity linking provides a strong support to generate high-quality candidate subjects.

In SimpleQuestions benchmark, a question, such as "which release was desperado the release track off of?", asks a direct relation of an entity called "desperado". While there are dozens of entities named "desperado" in Freebase which linked to different types and predicates[5]. Previous work either conduct relation inference firstly (Dai et al., 2016), or conduct entity linking firstly (Yih et al., 2015) may lead to no recall problem. In our framework, we propose to do joint fact selection to alleviate the problem. We leverage entities' name information and type information to represent entities' different aspects. As for predicates, we use the unique relation name and dispersive words information. In order to represent the sentence utterance properly, char-level and word-level encodings both are incorporated. The experimental results demonstrate the effectiveness of the proposed approach.

To sum up, our main contributions are: (1) We propose to conduct pattern extraction and entity linking, and put forward pattern revising procedure to mitigate the error propagation problem. (2) In order to learn to rank candidate subject-predicate pairs to enable the relevant facts retrieval given a question, we propose to do joint fact selection enhanced by relation detection. Multi-level encodings and multi-dimension information are leveraged to strengthen the whole procedure. (3) Our approach sets a new record in Simple KB-QA task, outperforming the current state-of-the-art by an absolute large margin.

[4]The question pattern of a question is produced by substituting the possible entity mention with a special token "#head_entity#" in original question. For example, for the question "*what position does carlos gomez play*", the correct entity mention is "*carlos gomez*", and the pattern should be "*what position does #head_entity# play*".

[5]Some predicate sets linked to these entities which has the same name "desperado", contains the gold predicate "music/release_track/recording" in common.

2 Related Work

The research of KB-QA has evolved from earlier domain-specific QA (Zelle and Mooney, 1996; Tang et al., 2001) to open-domain QA based on large-scale KB. There are two mainstream research directions for the KB-QA task. One of the promising approaches is semantic parsing (Cai and Yates, 2013; Yih et al., 2015; Yih et al., 2016; Reddy et al., 2016), which uses logic language CCG (zettlemoyer and Collins, 2009; zettlemoyer and Collins, 2012; Kwiatkowski et al., 2013; Reddy et al., 2014; Choi et al., 2015) or DCS (Berant et al., 2013) to map a question to its formal logical form to query on a KB. The other category exploit vector space embedding approaches (Bordes et al., 2014a; Bordes et al., 2014b; Bordes et al., 2015; Dong et al., 2015; Xu et al., 2016a; Xu et al., 2016b; Hao et al., 2017; Lukovnikov et al., 2017) to measure the semantic similarity between question utterance and candidate resources in the background KB, such that the correct supporting evidence will be the nearest neighbor of the question utterance in the learned vector space.

Instead of measuring the similarity between a question and an evidence triple with a single model, Yih et al. (2015) adopt a multi-stage approach. In each stage, one element of the triple is compared with the question utterance to produce a partial similarity score by a dedicated model. Then these partial scores are combined to generate the overall measurement condition. Dong et al. (2015) use three columns of CNNs to represent questions respectively when dealing with different answer aspects. Xu et al. (2016a; 2016b) incorporate Wikipedia free text to address KB-QA problems, in which they use multi-channel CNNs to extract relations. Dai et al. (2016) employ a conditional factoid factorization by inferring the target relation first and then the target subject associated with the candidate relations. Yin et al. (2016b) stack an attentive maxpooling above convolution layer to model the match of candidate predicates and questions. Lukovnikov et al. (2017) train a neural network, which contains a nested word/character-level question encoder, for answering simple questions in an end-to-end manner. In this work, we propose a two-stage framework, which exploit multi-dimension information leveraging multi-level encodings to compute semantic similarity, to tackle the problem of simple KB-QA.

Relation detection for KB-QA starts with feature-rich approaches (Yao and Van Durme, 2014) towards usages of vector space embedding models (Yih et al., 2016; Xu et al., 2016a; Golub and He, 2016; Yu et al., 2017). Actually many of the mentioned researches could support large relation dictionary, fitting the goal of open-domain question answering. Some work (Yin et al., 2016b; Yu et al., 2017) split relations into word sequences for single-relation detection. Golub and He (2016) propose a generative model for relation detection which predicts relation in a character-level sequence-to-sequence manner. Many researches have proved that relation detection benefit for KB-QA task. In our approach, we use relation detection enhance entity linking results, generating a more focused candidate subject set.

Conducting fact selection in KB-QA is inspired by work on answer selection (Yu et al., 2014; Yin et al., 2016b) which looks for correct answers from some candidates given a question. In machine comprehension (Yin et al., 2016a) scenario, the answer candidates are raw text, not structured information as facts in KB are. In our joint fact selection procedure, we leverage multi-level encodings of question utterance and subject-predicate pairs to measure their similarity in order to get the retrieved answer.

3 Overview of Simple Question Answering

The goal of simple KB-QA task could be formulated as follows. Given a natural language question q, the system returns the answer $(\hat{s}, \hat{p})$. The architecture of our proposed framework is shown in Figure 1, which illustrates the basic flow of our approach. In our approach, we conduct pattern-revising enhanced pattern extraction and entity linking to identify each question's corresponding pattern and mention, and generate candidate subjects together with their corresponding candidate facts from Freebase. Then a multi-level encoding based deep neural network is employed to select fact from the generated candidate facts, i.e. the joint fact selection procedure, in which relation detection is leveraged to reorder the entity linking results. The similarity score between the question utterance and each corresponding candidate fact is calculated, and the candidate fact with the highest score will be selected as the final predicted answer to the question.

Freebase (Bollacker et al., 2008) is utilized as our background KB, which has more than 3 billion facts,

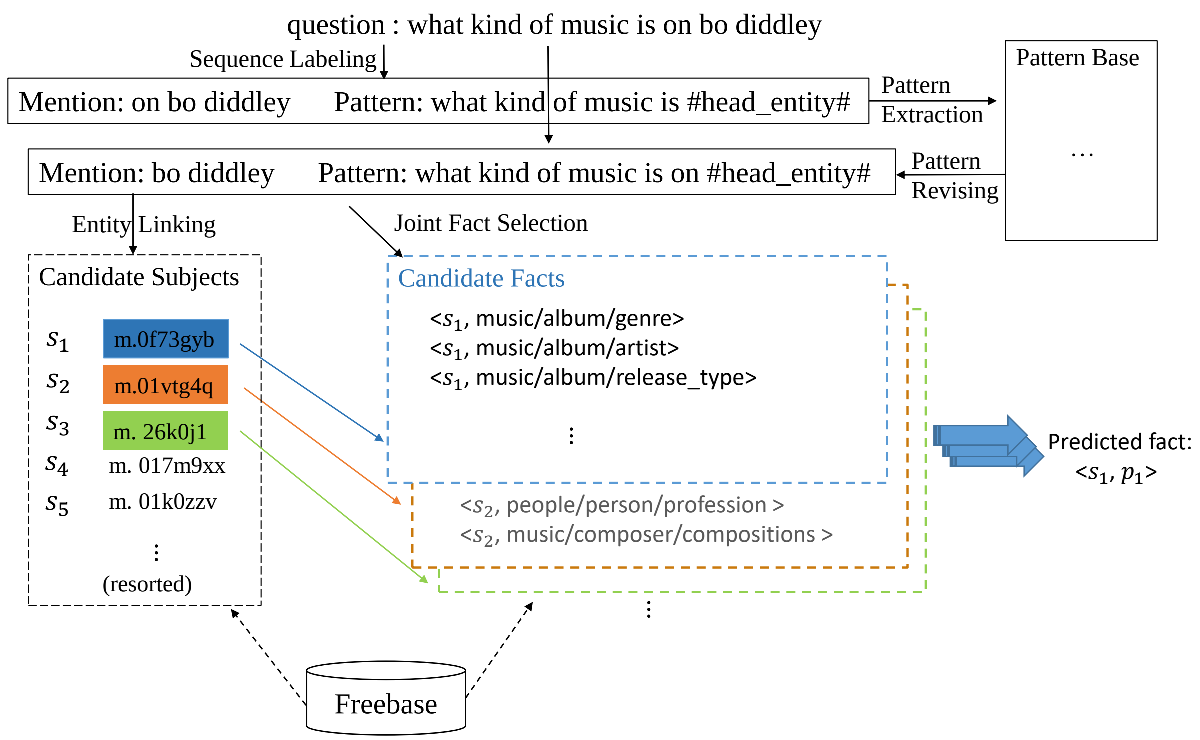

Figure 1: The overview of our proposed simple question answering approach, which is tailored via first stage of *pattern extraction and entity linking* enhanced by pattern revising, and second stage of *joint fact selection* which learns to rank candidate subject-predicate pairs to enable the relevant facts retrieval given a question.

and is used as the supporting KB for many QA tasks. We use SimpleQuestions dataset as our benchmark. The SimpleQuestions dataset consists of more than one hundred thousand simple questions which can be answered with a single fact in Freebase. We will give more detail in Section 6.1.

4 Pattern Extraction and Entity Linking

Given a question, two problems should be carefully tackled: (1) identifying the mention span in the question that refers to an entity in KB, (2) and generating candidate entities in KB that the question should refers to. So we conduct pattern extraction and entity linking procedure, which is enhanced by pattern revising. This is the first stage of our approach.

Given a question, the first stage should provide a set of top-N subject candidates. Generally, we should use all N-grams to retrieve the candidate subjects in KB in order to guarantee the recall rate. In this case, the candidate space is restricted to entities whose name or alias matches an n-gram of the question, as in (Bordes et al., 2015). However, the candidate pool is too noisy due to the large number of non-subject-mention n-grams. So we try to find a method to split the question utterance into two parts: mention span and question pattern. Mention span should be relatively rare words that refers to an entity in KB, while question pattern usually consists of common words that reflect the relation the question utterance refers to.

Inspired by previous work (Dai et al., 2016; Yin et al., 2016b), the problem could be treated as a sequential labeling task in which we aim to label the relatively rare continuous words of mention span to one kind, and label the relatively common words which belong to question pattern to another kind. This is the question pattern extraction step. The key idea is to train a model to predict the continuous word of text mention span which may refers to the topic entity, similar as Name Entity Recognition (NER). In the training set of SimpleQuestions, the topic entity of each question is labeled. We map the gold entity back to the text to label the text mention span for each question, then train a BiLSTM-CRF model to detect the subject entity mention span in question utterance. After substituting each subject entity mention span by "#head_entity#", we get the question pattern for each question. Each question is split into (mention, pattern) pair. The extracted patterns are taken in the pattern base.

4.1 Pattern Revising

Due to the performance of sequential labeling model, there are a certain portion of bad cases in our (mention, pattern) pairs result. The more common kind of bad patterns are the ones that have obvious mistakes together with the wrong labeled mention span. The more extreme error type is that, for instance, mention is the question utterance and pattern is null. Based on the principle that question patterns which consist of common words should occur in dataset more than one time and the more times a pattern appears the more likely it is to be correct, we propose a novel pattern revising procedure to correct wrong patterns in order to increase the proportion of "good patterns", following the steps illustrated in Algorithm 1[6]. We keep all the extracted pattern which appears more than one time as the pattern base $P_{n'}$, and all the original questions as the sentence utterance base Q_n. For each question in the sentence utterance base, we reappraise its extracted pattern and try to find whether we can find a more suitable pattern or not.

Algorithm 1 Pattern Revising

Input:

The set of extracted pattern base, $P_{n'}$;
The set of original question utterance base, Q_n;

Output:

The set of the revised Question-Pattern base, Q_P;

for each sentence utterance $q \in Q_n$ **do**
  for each candidate pattern $p \in P_{n'}$ **do**
    split p according to "#head_entity#";
    if all split results are contained in q and p, q satisfy consistency policy **then**
      add p to candidate pattern set q_{cp}
    end if
  end for
  if $|q_{cp}| == 1$ **then**
    set $p \in q_{cp}$ as pattern q_p
  else
    calculate word match count for each $p \in q_{cp}$
    select p with max word match count as pattern q_p
  end if
  add (q, q_p) to Q_P
end for

After getting all the revised pattern, we extract their corresponding mention span for each question utterance. The mention span is used to link entities in KB. Based on the mention span, we use each word of it to retrieve subject entities whose names contain this word, deriving the longest consecutive common subsequence for each candidate subject entity. Then we rank the candidate subject entities according to each entity's longest consecutive common subsequence, similarly as Yin et al. (2016b) did. Top-M ranked entities are kept for each question, denoted as C'_e, and their scores for ranking are denoted as $s_{linker}(e, q)$.

[6]The consistency policy means that the split results combine the words which are in the position of "#head_entity#" should equal to the original question utterance. It is not allowed to have redundant words.

5 Joint Fact Selection

5.1 Relation Detection enhanced Subject Candidates

For one question, there may be a large set of candidate entities. It's unrealistic to use all of them to generate candidate facts in practice. We need to let the candidate entities which are more likely to generate the gold fact get ahead of the rest. So we conduct relation detection to resort the subject candidates. We train a relation detection network which maps each question utterance to their most possible predicate. Predicates are represented from different granularity: predicate-level and word-level. Predicate-level is represented by the predicate name as a single token, while word-level is represented by splitting predicate name into word sequence. Word-level encoding serves as a supplement to predicate-level encoding, mitigating data sparsity[7].

We use question utterance as input for a relation detector to score all predicates connected to entities in C'_e, for each question. The similarity score $s_{rel}(r,q)$ is computed using cosine distance between the LSTM-based encoding of question utterance and the predicate embedding r_{pred}. We extract candidate subjects' notable type on behalf of candidate subjects' type information. Then word-level LSTM-based encoder is leveraged to get the type information embedding:

$$r_{type} = ENC_{type}(w_1, w_2, ...) \tag{2}$$

We represent predicate from different granularity: predicate-level and word-level, jointly considering the corresponding candidate subject's type information:

$$r_{pred} = [e_{pred-lev}; ENC_{word-lev}(w_1, w_2, ...); r_{type}] \tag{3}$$

For each entity $e \in C'_e$ and its associated relation R_e, we generate the final rank score:

$$s_{rank}(e,q) = s_{linker}(e,q) + \alpha(\max_{r \in R_e} s_{rel}(r,q) + \beta) \tag{4}$$

where α and β are constant hyperparameters. Finally top-N ($N < M$) ranked entities are kept for each question to generate fact pool, denoted as $\mathcal{C}_s$.

For each candidate subject entity s in $\mathcal{C}_s$ and its associated predicate set P_s, we can generate candidate fact pairs (s,p) ($p \in P_s$). Then for each question, we have all candidate fact pair pool $\mathcal{C}_{(s,p)}$. From a global perspective, the last is fact selection problem, also a matching task. Given a question q $(mention, pattern)$ and sets of candidate subject entities and predicates, $\mathcal{C}_s = \{s_1, ..., s_n\}$ and $\mathcal{C}_p = \{p_1, ..., p_m\}$ respectively, our fact selection model should jointly returns the subject and predicate that matches the question best. The calculation process is shown in the Figure 2.

5.2 Subject Matching Network

Entity names are used to calculate similarity score with the mention span detected in question utterance. Since the coverage of word embeddings is limited, many words in entity names are out-of-vocabulary (OOV) words, so we leverage character-level RNN-based encoding (the string is seen as a sequence of characters) for mention span and candidate subjects' names, mitigating the high prevalence of OOV problem.

As illustrated in Figure 2, we first look up a character embedding matrix E_c, which is randomly initialized and updated during the training process, to get the character embeddings for the entity name and the mention span. Then the embeddings are fed into an unidirectional LSTM networks, where the last hidden state is taken as the encoding of current sequences. Then we compute cosine similarity $t_{subj}(s,q)$ between the encodings of the entity name and the mention span in string surface-form.

[7]There are a certain portion of predicates' names not appearing in training data.

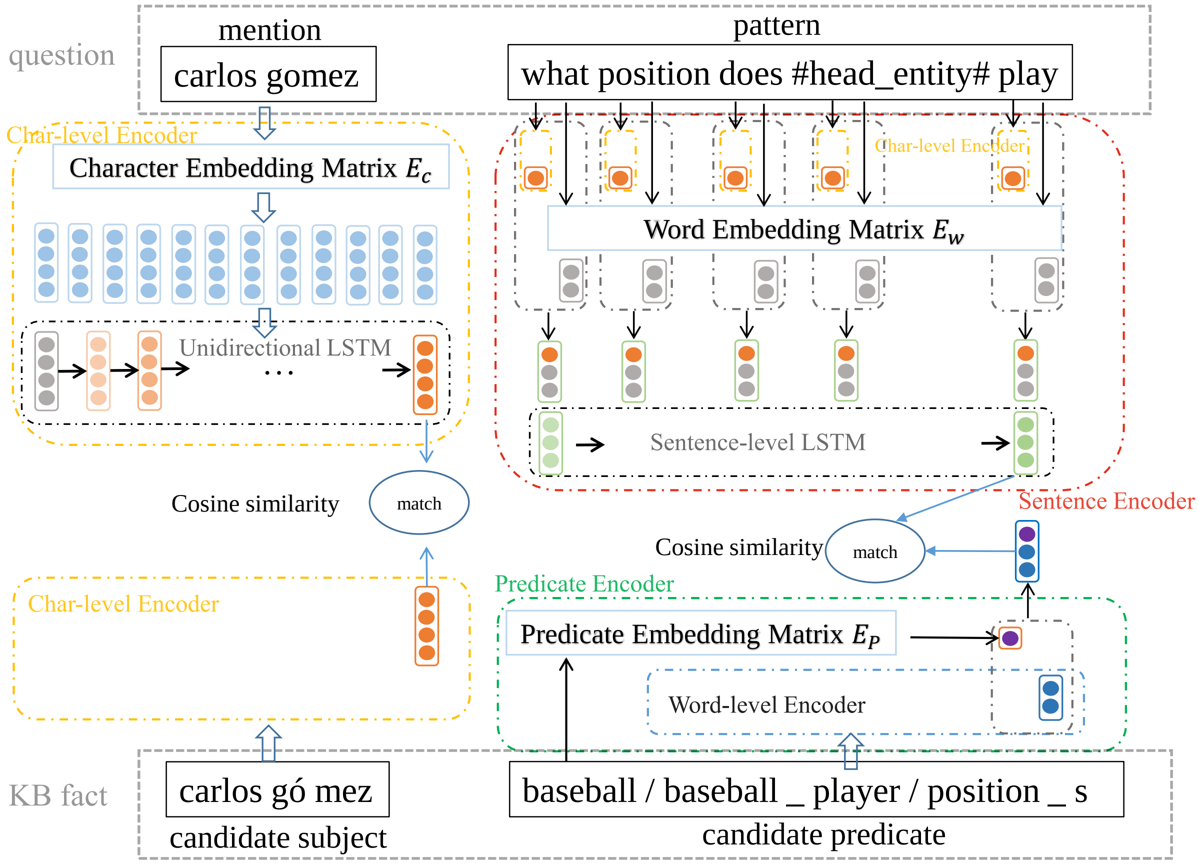

Figure 2: The overview of the proposed fact selection procedure. The mention span in the question utterance and the name of candidate subject are encoded using char-level encoder, then cosine distance is calculated to represent their similarity score. The sentence encoder for question pattern is in a nested structure, both leverage each word's char-level encoding and word embedding. The predicate encoder is similar as in 5.1, utilizing predicate-level encoding and word-level encoding.

5.3 Predicate Matching Network

Predicate matching network tries to match the patterns extracted from question utterance with the corresponding candidate predicates.

[**Question pattern encoder**] The question pattern encoder maps the question pattern to its corresponding predicate in the vector space. The process is done by using a LSTM based encoder network.

As for word representation in pattern utterance, we exploit a nested hierarchical encoding manner, leveraging both character-level and word-level information of each word in question pattern. Char-level encoder is first employed to get the representation for each word, which is a microcosmic operation. Then the microcosmic encoding on character level is concatenated with the macroscopical word embedding, which is randomly initialized and updated during the training process, to get the representation of each word:

$$m_i = [e_{w_i}; ENC_{char-lev}(c_1, ..., c_t)] \tag{5}$$

This is the way how we get the nested word embeddings. For sentence level encoding, we use word-level LSTM encoder network to get question pattern representation:

$$r_{pattern} = ENC_{pattern}(m_1, ..., m_n) \tag{6}$$

[**Predicate encoder**] In the matching procedure, we incorporate the candidate predicate's hierarchical encoding information to measure similarity with question pattern. This is similar with what we did in Section 5.1.

We represent predicate from different granularity: predicate-level and word-level. In detail, the microcosmic encoding on word-level is done by using a word-level LSTM based encoder network. Then the encoding is concatenated with its upper macro predicate-level encoding:

$$r_{pred} = [e_{pred-lev}; ENC_{word-lev}(w_1, w_2, ...)] \tag{7}$$

Both predicate-level and word-level encodings are useful to represent the predicate properly. We compute cosine similarity between r_{pred} and $r_{pattern}$ to get the matching score $t_{pred}((s,p),q)$.

5.4 Training

The overall ranking score of a fact triple t is $S(t,q) = t_{subj}(s,q) + t_{pred}((s,p),q)$. Our objective is to minimize the ranking loss:

$$L_{q,t,t'} = [\gamma + S(t',q) - S(t,q)]_{+} \tag{8}$$

where γ is a hyper-parameter, t' is a negative triple. Our fact pool consists of all facts whose subject entity is in the top-N entity candidates. For train, we sample 200 negative facts for each ground truth fact; and keep all candidate facts for valid and test.

6 Experiments

6.1 Dataset and Evaluation

SimpleQuestions benchmark is a typical Simple QA task, which provides a set of simple questions that can be answered by a single Freebase triple[8]. This dataset is split into three parts: train (75,910 instances), valid (10,845 instances) and test (21,687 instances) sets. The benchmark also provides two subsets of Freebase: FB2M (2,150,604 entities, 6,701 predicates, 14,180,937 atomic fact triples), FB5M (4,904,397 entities, 7,523 predicates, 22,441,880 atomic fact triples).

The evaluation metric is accuracy. Only a fact that matches the ground truth answer in both subject and predicate is counted as correct.

$$accuracy = \frac{\sum_{i=1}^{N} 1_{[(\hat{s}_i,\hat{p}_i)=(s_i,p_i)]}}{N} \tag{9}$$

6.2 Experimental Settings

In our approach, three types of embedding matrices are utilized: character embedding matrix, word embedding matrix, and predicate-level KB recource embedding matrix (i.e., the predicate embedding matrix). All of them are randomly initialized and updated during training process. The dimension of character embedding and predicate embedding is set to 100, and 200 for word embedding matrix. The hidden layer size of LSTMs is consistent to their corresponding dimension of the sequences to be processed. The candidate subjects number N is 20. We employ Adagrad (Duchi et al., 2011) to minimize the hinge training loss, where the margin γ is set to 1.0, and negative sample number is 200. All hyperparameters are determined according to the performance on the validation set.

6.3 Results

To demonstrate the effectiveness of the proposed approach, we compare our method with state-of-the-art systems.

Bordes et al. (2015) proposed an implementation of memory network for SimpleQuestions task. Golub and He (2016) tackled Simple QA using a character-level attention-based encoder-decoder LSTM model. A RNN-based approach was proposed by Dai et al. (2016), which utilized a focused pruning method called CFO. Yin et al. (2016b) leveraged an attentive CNN model to deal with the problem of Simple QA, and achieved best experimental result in previous state-of-the-art systems. Lukovnikov et al. (2017) adopted an end-to-end neural network, achieving a relatively good result.

[8]The object entity in the fact triple is the answer to the question utterance.

Approach	Setting	$Accuracy$
Dai et al. (2016)	N-Gram+	62.6*
Bordes et al. (2015)	end-to-end	62.7
Yin et al. (2016b)	passive linker	68.3
Golub and He (2016)	end-to-end	70.9
Lukovnikov et al. (2017)	end-to-end	71.2
Dai et al. (2016)	focused pruning	75.7*
Yin et al. (2016b)	attentive pooling	76.4
Our approach	**pattern-revising + joint fact selection**	**80.2**

Table 1: Comparison with state-of-the-art systems on SimpleQuestions benchmark. When marked with (*), accuracy in FB5M setting is given, otherwise FB2M setting results are shown.

The experimental results demonstrate that our framework sets new record in this task, outperforming the the state-of-the-art by an absolute large margin, achieving a competitive result, as shown in Table 1. From the results, we observe that our framework outperforms the latest end-to-end system record of (Lukovnikov et al., 2017) by 9.0 points, higher than previous best reported system result in (Yin et al., 2016b) by 3.8 points. While our framework exploits a quite simple and universal model which can be easily applied to other simple KB-QA tasks, not using attention model or other complex settings.

6.4 Framework Analysis

In this part, we further discuss the impacts of the augmented components which can be removed in our approach. Table 2 indicates the effectiveness of different parts in the framework.

Approach	Setting	$Accuracy$
our approach	fully integrated	80.2
our approach	w/o pattern revising	77.8
our approach	w/o relation detection	77.5
our approach	w/o both	74.8

Table 2: The ablation results of our approach.

The accuracy of original system is 74.8%. When augmented with pattern revising procedure, the accuracy of our framework increase 2.7 points, indicating that our pattern revising algorithm does lift the quality of generated patterns. The accuracy increase to 77.8 when we use relation detection to resort the results of subject candidates, showing that incorporating predicate and entity type information for reranking candidate entities can promote the quality of the recalled candidates. Either component is important for our framework. The performance decreases obviously when we remove any of them.

6.5 Error Analysis

We perform error analysis on our approach by randomly sampling 100 wrongly answered questions. The errors can be categorized into 4 types.

The most common kind of error (56%) is the wrongly picked subject entity or predicate. Due to the performance of pattern extraction and entity linking procedure, the gold entity may not appear in our candidate subjects. In this case our joint fact selection process can not select the gold subject entity. The probability of selecting wrong subjects is much higher than selecting a wrong predicate. Another kind of error case (22%) could be concluded into the “long drug name” error[9]. Some drug entities’ names are quite long (more than 15 words), while mention span in question utterance may be much shorter (less than 4 words). In this case our mention-subject matching network in the joint fact selection process is failed. The third common error (13%) can be classified as “KB error”. There are some entities with the

[9]For example, in the question “what is an active ingredient in childrens earache relief”, the labeled gold subject “m.0jxn044” has a name longer than 15 words.

same name and the same type, indeed. The KB or the question can't support more evidence to distinguish them. The last type of error (9%) is label error. In the dataset there is a small portion of questions which are repeated, but the labeled subject or predicate is different.

7 Conclusion

In this work, we propose a pattern revising procedure to mitigate the error propagation problem in pattern extraction and entity linking process. In order to learn to rank candidate subject-predicate pairs to enable the relevant facts retrieval given a question, we propose to do joint fact selection enhanced by relation detection. Multi-level encodings and multi-dimension information are leveraged to strengthen the whole procedure. The experimental results demonstrate the effectiveness of the proposed approach.

Acknowledgements

The research work is supported by the National Key Research and Development Program of China under Grant No.2017YFB1002101, the Natural Science Foundation of China (No.61533018 and No.61702512), and the independent research project of National Laboratory of Pattern Recognition. This work is also supported in part by Beijing Unisound Information Technology Co., Ltd.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. *Semantic Parsing on Freebase from Question-Answer Pairs*. EMNLP 2013, volume 2.

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

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. *Semantic Parsing on Freebase from Question-Answer Pairs*. arXiv preprint arXiv:1406.3676.

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

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. *Open question answering with weakly supervised embedding models*. Joint European Conference on Machine Learning and Knowledge Discovery in Databases:165–180. Springer.

Qingqing Cai, Alexander Yates. 2013. *Large-scale Semantic Parsing via Schema Matching and Lexicon Extension*. ACL 2013:423–433.

Eunsol Choi, Tom Kwiatkowski, and Luke Zettlemoyer. 2015. *Scalable Semantic Parsing with Partial Ontologies*. 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):1311–1320.

Zihang Dai, Lei Li, and Wei Xu. 2016. *CFO: Conditional Focused Neural Question Answering with Large-scale Knowledge Bases*. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers):800–810.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. *Question Answering over Freebase with Multi-Column Convolutional Neural Networks*. Proceedings of the 53th Annual Meeting of the Association for Computational Linguistics:260–269.

Join Duchi, Elad Hazan, and Yoram Singer. 2011. *Adaptive subgradient methods for online learning and stochastic optimization*. JMLR, 12:2121–2159.

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*. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers):221–231.

David Golub, and Xiaodong He. 2016. *Character-Level Question Answering with Attention*. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing:1598–1607.

Sepp Hochreiter, and Jrgen Schmidhuber. 1997. *Long short-term memory*. Neural computation 9(8):1735–1780.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. *Scaling semantic parsers with on-the-fly ontology matching*. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.

Denis Lukovnikov, Asja Fischer, and Jens Lehmann. 2017. *Neural Network-based Question Answering over Knowledge Graphs on Word and Character Level*. International Conference on World Wide Web 2017:1211–1220.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. *Large-scale Semantic Parsing without Question-Answer Pairs*. Transactions of the Association of Computational Linguistics, volume 2:377–392.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. *Transforming dependency structures to logical forms for semantic parsing*. Transactions of the Association for Computational Linguistics, volume 4:127–140.

Lappoon R. Tang, and Raymond J. Mooney. 2001. *Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing*. European Conference on Machine Learning 2001:466-477.

Christina Unger, André Freitas, and Philipp Cimiano. 2014. *An introduction to question answering over linked data*. Reasoning Web International Summer School:100-140. Springer.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. *Hybrid Question Answering over Knowledge Base and Free Text*. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers:2397–2407.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. *Question Answering on Freebase via Relation Extraction and Textual Evidence*. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics:2326–2336.

Xuchen Yao, and Benjamin Van Durme. 2014. *Information Extraction over Structured Data: Question Answering with Freebase*. Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics:956–966.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. *Semantic Parsing for Single-Relation Question Answering*. Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics:643–648.

Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. *Semantic parsing via staged query graph generation: Question answering with knowledge base*. Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. *The value of semantic parse labeling for knowledge base question answering*. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.

Wenpeng Yin, Sebastian Ebert, and Hinrich Schütze. 2016. *Attention-based convolutional neural network for machine comprehension*. Proceedings of NAACL Human-Computer QA Workshop.

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. *Simple Question Answering by Attentive Convolutional Neural Network*. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers:1746–1756.

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*. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers):571–581.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. *Deep learning for answer sentence selection*. Proceedings of 2014 ICLR Workshop.

John M Zelle, and Raymond J Mooney. 1996. *Learning to parse database queries using inductive logic programming*. Thirteenth National Conference on Artificial Intelligence:1050–1055.

Luke S Zettlemoyer, and Michael Collins. 2009. *Learning context-dependent mappings from sentences to logical form*. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP:976–984.

Luke S Zettlemoyer, and Michael Collins. 2012. *Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars*. arXiv preprint arXiv:1207.1420.