2017 10th International Symposium on Computational Intelligence and Design

Open Domain Question Answering with Character-level Deep Learning Models

Kai Lei, Yang Deng, Bing Zhang, and Ying Shen*
Institute of Big Data Technologies
Shenzhen Key Lab for Cloud Computing Technology & Applications
School of Electronic and Computer Engineering, Peking University
SHENZHEN 518055, P.R. China
leik@pkusz.edu.cn, {ydeng, zhang_bing}@pku.edu.cn, shenying@pkusz.edu.cn

***Abstract*—Single-relation factoid question answering (QA) is strongly supported by rich sources of facts from knowledge bases (KB). However, there are many irrelevant information in questions and overwhelming number of facts in knowledge bases, making it difficult to capture goal entity and relation involved in a question. In order to settle these issues, firstly, a state-of-the-art sequence tagging model (BiLSTM-CRF) is adopted to detect the entity mention in a question. Then, we propose a n-gram match (NGM) algorithm with Chinese-specific rules and an attention-based siamese bidirectional long-short term memory (ASBLSTM) model to measure the lexical and semantic similarity between questions and candidate facts. Our whole method requires no hand-crafted template or feature engineering. In addition, character-level models are proved to be effective in solving the out of vocabulary (OOV) issue and improving the accuracy in Chinese KBQA task. Experiment results show that our system outperforms the best system with deep learning models in the KBQA share task of the Conference on Natural Language Processing and Chinese Computing (NLPCC) 2016 and our system achieves an AverageF1 measure of 80.97% and 37.18% on test dataset in NLPCC 2016 and 2017 respectively.**

Keywords—Knowledge Base Question Answering, Entity Extraction, Relation Recognition, Entity Linking.

I. INTRODUCTION

Single-relation factoid question answering (QA) is the most demanding area in open-domain QA system. In recent years, several efforts have been made on constructing large-scale knowledge bases (KB), such as Freebase [1] and DBPedia [2]. The development of KB leads to a boom of researches on KBQA. Most of KBQA researches are experimented on data sets in English, such as WebQuestions [3] and SimpleQuestions [4]. In the field of Chinese KBQA research, the KBQA share task of the Conference on Natural Language Processing and Chinese Computing (NLPCC) 2016 [5] provides the first large-scale Chinese KBQA data set, and a new test set is released in 2017. In this paper, we focus on solving the single-relation factoid KBQA issue in Chinese, based on the NLPCC data set.

In KBQA, the goal is to map a question to a single fact (subject, predicate, object) in KB. For example, in the question of "Who is the author of Harry Potter?", the subject and the predicate are supposed to be "Harry Potter" and "Author" respectively. Then we retrieve the fact (Harry Potter ||| Author ||| J.K. Rowling) from KB. In this procedure, there are several key issues remained to be settled. First, there are many noises in a question, making it difficult to extract the topic entity. Second, the knowledge base has an overwhelming number of facts. Third, it is almost impossible to cover all entities in training time. In this paper, we propose a complete method to overcome these obstacles.

First of all, we use a bidirectional long short-term memory model combined with conditional random field (BiLSTM-CRF) [6] to prune a question to an entity mention, which reduces redundant information in the question to a great extent. Second, an N-gram matching algorithm is designed to compute lexical similarity between the entity mention and the subject in KB, and also prepare the candidate facts, while an attention-based Siamese Bi-LSTM model (ASBLSTM) is constructed to measure semantic similarity between the question pattern and predicate candidates. Third, both BiLSTM-CRF and ASBLSTM are character-level models, which settles the OOV (out of vocabulary) problem in a decent way. Finally, final answers are selected by re-ranking the score of subject and predicate candidates. Our overall deep learning based approach requires neither any handmade template nor feature engineering, and also outperforms the best result with a deep learning based model.

The rest of our paper is structured as follows: Section 2 discusses related work, Section 3 gives a detailed description of the overall framework of our method, Section 4 presents experimental setup, results and analysis, and Section 5 summarizes this work and the future direction.

II. RELATED WORK

Open-domain QA system used to be designed as a search engine-based system, which utilizes keywords extracted from the question to retrieve evidences from a search engine, and the answer is generated from those evidences [7,8].

Recently, with the development of large-scale knowledge bases, KB-based QA becomes a main-stream approach to solve open-domain QA task. Most approaches are based on semantic parsing [9,10]. Besides, state-of-the-art results are mostly achieved by deep learning based approaches [5,11,12].

Our approach is motivated by the idea of deep learning method as mentioned above, character-level language modeling and attention mechanism. Compared with word-level modeling, character modeling is proved to efficiently handle English OOV issue not only in QA task [13], but also in most of NLP task. Attention mechanism also improves the result in QA [13, 14] as well as other NLP task.

In the latest benchmark NLPCC 2016, Lai et al. [15] proposed a template-based SPE (subject predicate extraction) algorithm, which achieved the state-of-the-art result, F1-score of 82.47%. However, the proposed QA system requires considerable hand-crafted templates, which is not only time-consuming but also unable to be fit in different data set.

2473-3547/17 $31.00 © 2017 IEEE
DOI 10.1109/ISCID.2017.58

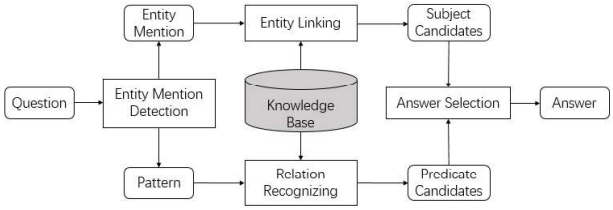

Figure. 1. Overview of the KBQA System Framework

III. METHOD

The overview of our KBQA system framework is shown in Fig. 1. As illustrated in Figure 1, our method consists of four modules:

(1) Entity Mention Detection Module. A character-level BiLSTM-CRF model for detecting the topic entity mentioned in a question.

(2) Entity Linking Module. An approach combined N-Gram Match with some Chinese-specific information for linking the entity mention to a set of candidate subjects from KB.

(3) Relation Recognition Module. A character-level Attention-based Siamese Bi-LSTM model (ASBLSTM) for recognizing the predicate in a question pattern.

(4) Answer Selection Module. An approach for re-ranking the candidate triples to select final answers.

The detail of each module are described as follows.

A. Entity Mention Detection

Given a question, firstly, we have to identify the entity mention in the question. This approach to mention detection produces a (mention, pattern) pair for each question. For example, with the input "《机械设计基础》这本书的作者是谁？", the mention detection model outputs a pair like ("机械设计基础", "《_》这本书的作者是谁？"), in which "_" indicates the topic entity in the question.

We trained a character-level BiLSTM-CRF model to detect the entity mentions. Since we use a character-level model, each question is labeled character-by-character under a "BMEO" annotation rule. For example, the labeled of the question is "《/O 机/B 械/M 设/M 计/M 基/M 础/E 》/O 这/O 本/O 书/O 的/O 作/O 者/O 是/O 谁/O ？ /O". Besides, character embeddings are trained to represented the natural language questions rather than word embeddings. The vector representations are generated by pre-trained models.

As shown in **Figure. 2**, the LSTM layer contains two sub-networks for head-to-tail and tail-to-head contexts. The output of the i^{th} word is $h_i = [\overrightarrow{h_i} \oplus \overleftarrow{h_i}]$, in which $\overrightarrow{h_i}$ is the output of the forward network and $\overleftarrow{h_i}$ is that of the backward network.

The final layer of the model is a CRF layer to decode the entire output of the LSTM layer into a labeled sequence. The whole entity mention detection model is commonly referred to as BiLSTM-CRF [6], but in character-level, which is proved to outperform the model a word-level.

B. Entity Linking

Based on the detected mention, we need to link the mention to a specific entity from the knowledge base. Entity linking aims at generating the entity candidates for each entity mention via following steps.

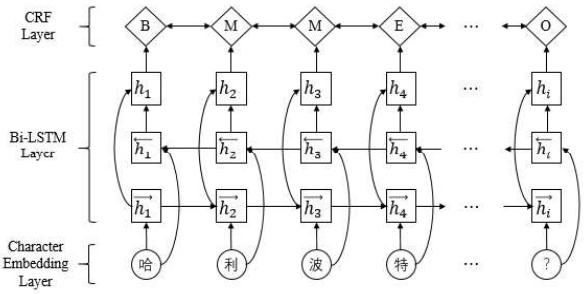

Figure. 2. The Architecture of Character-level BiLSTM-CRF Model

(1) Generate all the n-grams of the entity mention as $N = \{n_1, n_2, \ldots, n_k\}$

(2) Use each character of the entity mention to retrieve candidates which cover at least half of those characters

(3) We propose a N-Gram Matching(NGM) approach to measure the similarity between a mention $\boldsymbol{m}$ and a knowledge base entity $\boldsymbol{e_i}$

$$\boldsymbol{NGM(m, e_i) = \sum_{j=1}^{k} |n_j| * (n_j \& e_i)} \qquad (1)$$

(4) For each entity candidate $\boldsymbol{e}$, compute its NGM with the mention $\boldsymbol{m}$. Let $\boldsymbol{p}$ be the position of the first character of the entity candidate in $\boldsymbol{m}$. Let $\boldsymbol{q}$ be a flag that equal 1 if $\boldsymbol{e}$ is before the character "的" in $\boldsymbol{m}$, or equal 0. Finally, entity candidate $\boldsymbol{e}$ is scored by the equation below:

$$\boldsymbol{score(e) = a\left(\frac{NGM}{|e|}\right) + b\left(\frac{p}{|m|}\right) + cq} \qquad (2)$$

Parameters a, b, c are tuned on dev. Top-N ranked entities are kept for each question. Then we compute the likelihood score of each candidate subject by:

$$\boldsymbol{P(e_j) = \frac{\exp(score(e_j))}{\sum_{i=1}^{m} \exp(score(e_i))}} \qquad (3)$$

NGM gives a higher score to candidates that cover more n-grams of the entity mention. Compared with the longest consecutive common subsequence (LCCS) and the Levenshtein Distance, NGM performs better in entity linking with a character-level entity detection model.

Factor $\boldsymbol{p}$ and $\boldsymbol{q}$ indicate some characteristics of Chinese questions. First, the topic entity is more likely to appear close to the beginning of the question, which is opposite in English questions. For example, in "《哈利波特》的作者是谁？" / "Who is the author of Harry Potter?", it is obvious to find that the entity "哈利波特" is in the front part of a question in Chinese, while the entity "Harry Potter" is in the latter part of a question in English. Besides, the topic entity also probably appears before the character "的", which is similar to "of" in English.

For each question, this approach produces a set of candidate subjects with probability score from the knowledge base.

C. Relation Recognizing

The relation recognizing module is used to extract the predicate from the pattern of a question. The pattern is generated by the entity mention detection module. We propose an Attention-based Siamese Bi-LSTM model (ASBLSTM) to measures the semantic similarity between the question pattern and predicates in the knowledge base.

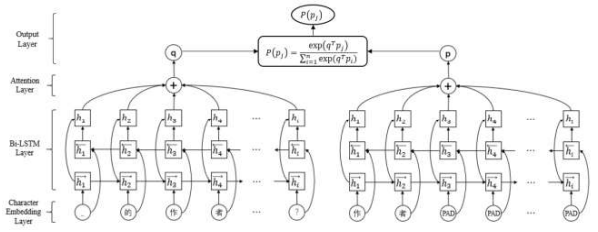

Figure. 3. The Architecture of Character-level ASBLSTM model

Our model is depicted in **Figure. 3**. There are a pair of attention-based Bi-LSTM network in our model, which is the same as the model Att-BLSTM [18], but in character-level. The question pattern and the candidate predicate are input into each network separately.

The whole ASBLSTM model contains five components in each Att-BLSTM model:

(1) Input layer: input sentence to the model;

(2) Embedding layer: map each character into a vector; same as the BiLSTM-CRF model in section 3.1;

(3) Bi-LSTM layer: use Bi-LSTM to get high level features from step (2); also same as the BiLSTM-CRF model in section 3.1;

(4) Attention layer: produce a weight vector, and merge character-level features from each time step into a sentence-level feature vector, by multiplying the weight vector; We now describe how we produce the sentence vector h_s. Let H be a matrix of vectors $[h_1, h_2, \dots, h_n]$ that the LSTM layer outputs, where n is the sentence length. The sentence vector h_s is computed by a weighted sum of these output vectors:

$$M = tanh(H) \tag{4}$$

$$\alpha = softmax(M\omega^T) \tag{5}$$

$$h_s = tanh(\alpha^T H) \tag{6}$$

where ω is a trained parameter vector and α is the attention distribution that is applied over each hidden unit, $H \in R^{n\times d^h}, \omega \in R^{1\times d^h}, \alpha \in R^{n\times 1}, h_s \in R^{1\times d^h}$.

(5) Output layer: multiply the output vector of two Att-BLSTM model to compute the semantic similarity score between the question pattern and the candidate predicate, and use softmax function to get a probability representation:

$$\boldsymbol{P(p_j) = \frac{\exp(q^T p_j)}{\sum_{i=1}^{n} \exp(q^T p_i)}} \tag{7}$$

D. Answer Selection

Given a set of subject candidates and a set of corresponding predicate candidates, {e} and {p}, we obtain the entity linking score of each subject candidate and the semantic similarity score of each predicate candidate from entity linking module and relation recognizing module respectively. Then we generate the most likely (subject, predicate) pair by:

$$\boldsymbol{(s, p) = \mathbf{argmax}_{e_i p_j}(P(e_i) * P(p_j))} \tag{8}$$

Finally, we extract (s, p, o) from the knowledge base, and the object becomes the final answer.

TABLE I. EXPERIMENTAL RESULTS IN NLPCC

Entity Detection	Relation Recognition	NLPCC2016	NLPCC2017
	C-DSSM	52.47%	-
TEEM	C-DSSM	78.08%	-
	BiLSTM-DSSM	75.29%	-
	BiLSTM-CNN-DSSM	78.15%	-
	Combined-DSSM	79.57%	-
BiLSTM-CRF+NGM	BiLSTM-DSSM	76.62%	34.48%
	Att-BLSTM	79.61%	36.04%
	ASBLSTM	**80.97%**	**37.18%**

IV. EXPERIMENT

A. Data Set

We train and evaluate our proposed method on the NLPCC 2016 and 2017 KBQA datasets. There are 14,609 questions with gold answers for training, and 9,870 questions for testing in NLPCC 2016 datasets. A new testing set of 48,850 questions without answers has been released in NLPCC 2017 for task evaluation. The training data is split into a training set (85%) and a validation set (15%). The knowledge base used to extract the answer is also provided. It includes about 6M subjects, 0.6M predicates, and 43M facts.

B. Experimental Settings

Both the character embeddings and the word embeddings are trained by Word2Vec [17], using a corpus of Chinese Wikipedia, and the embedding dimension is set to 100. For our model, we employ the same parameter settings in both BiLSTM-CRF model and ASBLSTM model. The size of hidden layers is set to 200. The learning rate and the dropout rate are set to 0.002 and 0.5 respectively. We train our models in batches with size of 128. The model parameters are regularized with a L2 regularization strength of 0.0001. Parameters a, b, c, discussed in section 3.2, are tuned to be 0.3, 1.0, 0.2 respectively. All other parameters are randomly initialized from [-0.1, 0.1]. The maximum number of characters in a sentence is set to 100 so that the model can cover all the questions in dataset.

C. Results

In this section, our proposed method is compared to two baseline methods, evaluated on the testing set from NLPCC 2016 and 2017.

Table 1 summarizes the experimental results of our method and two baseline methods. One is C-DSSM [19], provided in NLPCC 2016 as the benchmark systems. The other one is TEEM+Combined-DSSM [16], which is the best result in NLPCC 2016 by using a deep learning method without any feature and template engineering.

As is shown in the table 1, our method (BiLSTM-CRF+NGM+ASBLSTM) outperforming all other methods. With the same relation recognition model BiLSTM-DSSM, BiLSTM-CRF based entity detection method improves the average F1 by 1.33%. Other methods with BiLSTM-CRF based entity detection module also perform better than those with TEEM by 1.3% to 5.7%. This result indicates the superiority of BiLSTM-CRF in extracting the key information from a question. With the same entity detection model BiLSTM-CRF, attention-based relation recognition models

TABLE II. EXPERIMENTAL RESULTS WITH CHARACTER-LEVEL AND WORD-LEVEL MODELS

	Embed Type	Entity Mention Detection	Relation Recognition (F1@1)	KBQA
NLPCC 2016	Word	70.31%	49.35%	68.72%
	Char	87.01%	68.51%	80.97%
NLPCC 2017	Word	79.26%	-	29.86%
	Char	93.78%	-	37.18%

TABLE III. ANALYSIS OF CHARACTER-LEVEL AND WORD-LEVEL MODELS

	Embed Type	Entity Mention Detection		Relation Recognition	
		OOV %	#Vocab	OOV %	#Vocab
NLPCC 2016	Word	53.62%	20,820	62.9%	4,352
	Char	0.6%	1,721	0.2%	1,488
NLPCC 2017	Word	51.4%	-	63.8%	-
	Char	0.08%	-	0.05%	-

give better performance. Our proposed ASBLSTM model further improves the result from Att-BLSTM model.

Table 2 and Table 3 illustrate the effectiveness of our character-level methods to solve the OOV issue in KBQA task. In Table 2, we compare the same models with word-level embedings and character-level embeddings on both KBQA task and subtasks of entity mention detection and relation recognition, and Table 3 summarizes the percentage of OOV tokens and the vocabulary size in both character-level models and word-level models. The word-level model meets over 50% out-of-vocabulary words during training and testing, while the character-level model has barely no unseen character. These evidences clearly demonstrate that character-level models are able to settle the OOV issue in KBQA task effectively. As is presented in Table 2, methods using character-level models improve the f1 score by over 10% in every subtasks and different data set.

In addition, the vocabulary size in word-level models is 10 times and 3 times larger than that in character-level during subtasks of entity mention detection and relation recognition. Character-level models significantly trim the vocabulary so that the whole method turns out to be more efficient.

V. CONCLUSION

In this paper, we proposed a character-level KBQA system with deep learning models for single-fact question answering. We employ the BiLSTM-CRF model to reduces irrelevant information in entity mention detection, leading to significant improvement of overall accuracy. An N-gram match approach with Chinese-specific rules and an ASBLSTM model are designed to evaluate the degree of correlation and similarity between questions and facts. Finally, character-level models are proved to be both more effective and more efficient that word-level models in KBQA task. In the future, we will further explore how to better adopt linguistic knowledge. We consider how to combine data-driven model and priori knowledge model as the key part of QA in the future.

ACKNOWLEDGMENT

This work has been financially supported by Shenzhen Key Fundamental Research Projects (Grant No.: JCYJ20160330095313861, JCYJ20170412150946024 , JCYJ20170412151008290 and JCYJ20170306091556329).

REFERENCES

[1] Bollacker K, Evans C, Paritosh P, et al.: Freebase: a collaboratively created graph database for structuring human knowledge. In: ACM SIGMOD International Conference on Management of Data, pp. 1247-1250 (2008)

[2] Lehmann J, Isele R, Jakob M, et al.: DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. In: Semantic Web 6(2), 167-195 (2014)

[3] Berant J, Chou A, Frostig R, et al.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of EMNLP (2014)

[4] Bordes A, Usunier N, Chopra S, et al.: Large-scale Simple Question Answering with Memory Networks. In: Computer Science (2015)

[5] Duan N.: Overview of the NLPCC-ICCPOL 2016 Shared Task: Open Domain Chinese Question Answering (2016)

[6] Lample G, Ballesteros M, Subramanian S, et al.: Neural Architectures for Named Entity Recognition. In: The North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260-270 (2016)

[7] Yahya M, Berberich K, Elbassuoni S, et al.: Natural language questions for the web of data. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, pp. 379-390 (2012)

[8] Unger C, Lehmann J, Ngomo A C N, et al.: Template-based question answering over RDF data. In: International Conference on World Wide Web, pp.639-648 (2012)

[9] Cai Q, Yates A.: Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. In: Meeting of the Association for Computational Linguistics, pp. 423-433 (2013)

[10] Yao X, Durme B V.: Information Extraction over Structured Data: Question Answering with Freebase. In: Meeting of the Association for Computational Linguistics, pp. 956-966 (2014)

[11] Yih W T, Chang M W, He X, et al.: Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In: Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, pp. 1321-1331 (2015)

[12] Dai Z, Li L, Xu W.: CFO: Conditional Focused Neural Question Answering with Large-scale Knowledge Bases. In: Meeting of the Association for Computational Linguistics, pp. 800-810 (2016)

[13] Golub D, He X.: Character-Level Question Answering with Attention. In: Proceedings of EMNLP, pp. 1598-1607 (2016)

[14] Yin W, Yu M, Xiang B, et al.: Simple Question Answering by Attentive Convolutional Neural Network. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp. 1746-1756 (2016)

[15] Lai Y, Lin Y, Chen J, et al.: Open Domain Question Answering System Based on Knowledge Base. In NLPCC-ICCPOL (2016)

[16] Xie Z, Zeng Z, Zhou G, et al.: Knowledge Base Question Answering Based on Deep Learning Models. In NLPCC-ICCPOL (2016)

[17] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111-3119 (2013)

[18] Zhou P, Shi W, Tian J, et al.: Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In: Meeting of the Association for Computational Linguistics, pp. 207-212 (2016)

[19] Junwei B, Nan D, Ming Z, Tiejun Z.: Knowledge-based question answering as machine translation. In Proceedings of ACL (2014)