One Model Structure for All Sub-Tasks KBQA System

Hanchu Zhang, Deyi Xiong, and Xuanwei Nian

West Shangdi Road, Haidian District Beijing, China

Abstract. In this paper, we present a system that can answer natural language questions according to a Chinese knowledge base. Our system works in a pipeline manner where a semantic matching approach is proposed. We perform all sub-tasks by the same model structure with a few additional features. We build our model based on AllenNLP, and make the model publicly available on GitHub [1]. Our model can be reused as a baseline model for the future work.

Keywords: KBQA · Entity Linking · Semantic Matching.

1 Introduction

KBQA is a challenging task in natural language processing, of which the core issues include question understanding and representation, as well as detecting answers in a given knowledge base. In this paper, we introduce a system that can answer natural language questions according to a given knowledge base. Our system works in a pipeline manner and adopts a semantic matching approach. We accomplish all the sub-tasks by the same model structure. It recognizes topic mentions and then performs entity linking. We combine path ranking and answer finding into one task. After finding the first hop path and entities, we rank the 1-hop path and multi-hop path together. The trained model is able to distinguish the correct path and the corresponding answers at the same time. Our approach has achieved an F1 score 54.57% on CCKS-KBQA-2020 test dataset.

2 The Proposed System

In this section, we first describe the architecture of our system, and the workflow of the system, then introduce the neural network model used in the KBQA task.

2.1 The Architecture of the System

As shown in Figure 1, the system works in a pipeline manner. The workflow consists of two essential components: Topic Entity Recognition and Path Ranking.

In the Topic Entity Linking module, the system extracts the top entities from the input question utterance. The goal is to build a bridge between the question

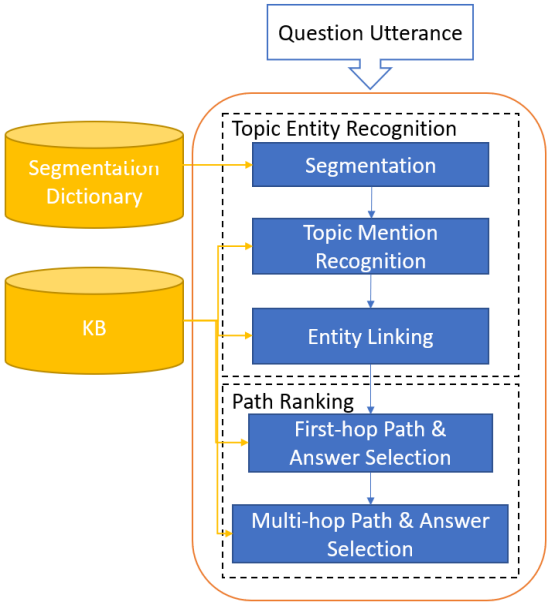

Fig. 1. Architecture of the proposed KBQA system.

and KB through topic entities, and do further processing in a more focused searching space. In the Path Ranking procedure, we combine path selection and answer selection into one task. We use the deep semantic matching model to select the path that best matches the question.

The first step is question utterance segmentation. The goal of this step is to detect all mentions as much as possible. The topic mention is the core concept in the question. Based on segmentation results, the Topic Mention Recognition sub-component then selects topic mentions from segments. The topic entity is the entity most concerned about in the question. The Entity Linking sub-component enumerates all the candidate entities in KB according to each mention, then performs entity disambiguation and links it back to KB, which outputs topic entities to the downstream tasks.

In the second step, the model takes all 1-hop paths of the topic entities as input, and finds the paths that match the utterance. Based on the previous step, the model continues to select the multi-hop paths that best match the utterance. Once we find the most suitable path, we also get the corresponding answers at the same time.

2.2 The Model

Problem Formulation We treat the KBQA as a combination of several semantic matching sub-tasks. We adopt the BiMPM [4] model for a series of matching tasks including: mention recognition, entity linking and path ranking. We make

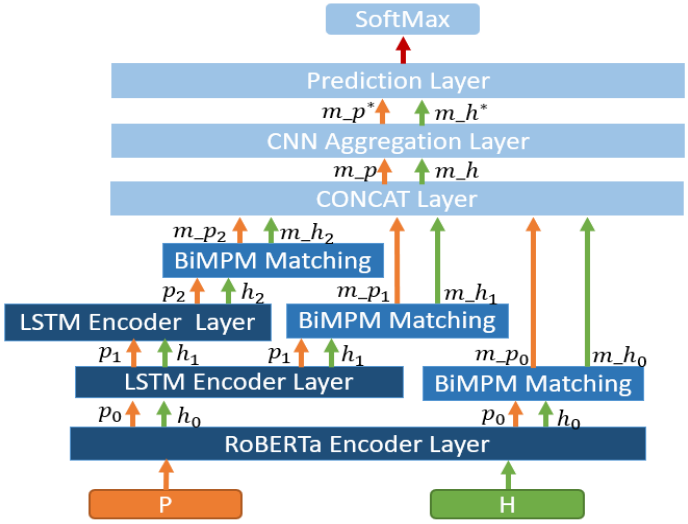

Fig. 2. Deep Semantic Matching Model

adaptations to the BiMPM so that every sub-task can obtain results through the same model structure. Given two sentences *Premise* and *Hypothesis*, the model learns whether the *Hypothesis* is inferable from the *Premise*. During prediction, we rank *Hypothesis* candidates according to their probabilities. If a *Hypothesis* whose score is higher than a threshold α, we keep it as a positive case otherwise discard it.

Deep Semantic Matching model As shown in Figure 2, our model first uses RoBERTa [3] to encode P into a sequence of contextualized token representations p_0, followed by two bidirectional LSTM layers to model further interactions between tokens, which produce representations p_1 and p_2. By the same way, we get three representations for H, namely h_0, h_1 and h_2. For each encoder layer's output embedding, we perform multi-perspective matching, and produce three matching vectors for each input text. Specifically, by doing matching operation on p_i and h_i, we obtain two multi-perspective matching vectors m_p_i and m_h_i, where $i \in \{0, 1, 2\}$.

The CNN Aggregation Layer is employed to aggregate vectors into two fixed-length matching vector for P and H. Specifically, We collect all the matching vectors of each input text and concatenate them into $\{m\_p_i\}$ and $\{m\_h_i\}$ respectively. For example, $concat(m\_p_0, m\_p_1, m\_p_2) = m\_p$. We then use a CNN layer to capture salient features of m_p into one compact matching vector m_p^*. The final matching vectors m_p^* and m_h^* are concatenated into one vector. The final prediction layer is a feedforward neural network with a single projection layer. After this final classifier we get the binary decisions for P and H pairs.

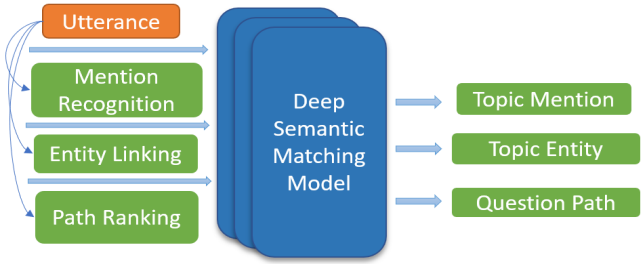

Fig. 3. Using the same model structure to implement different sub-tasks, including mention recognition, entity linking and path ranking.

Sub-Task Adaption Based on the above model structure, we adjust each sub-task as shown in Figure 3. For Topic Mention Recognition, we formulate *mention* as H. The Topic Mention Recognition is formulated as follows:

$$CandidateEntity[SEP]Predicate_1, Predicate_2, ..., Predicate_n \quad (1)$$

as H, where $Predicate_i$ is the 1-hop paths' predicate that surrounds the entity. The predicates surrounding the entity would enrich the context of the entity, and give more information about the entity to the model for disambiguation.

2.3 Path Ranking Strategy

Question utterance and path respectively represent two sentences that need matching in the model. The conventional construction method combines subject, predicate, and object to form a path. This method has two problems: a) As it is difficult to detect where the answer is, we usually need extra steps or precisely topic entity recognition result to find the answer, which may cause errors. b) The model is essentially semantic matching, introducing answers would introduce noisy to the model.

In order to solve the aforementioned problems, we propose an output path ranking strategy shown in Figure 4. Different from using $CandidatePath$ and $Question$ to do the matching, $PathforMatching$ and $Question$ have more obvious characteristics for the model to make the right decision. Considering the model has strong fitting ability, we dig out the corresponding answer from the candidate path. If we dig out entity from different locations in one path, the detected path could produce multiple path strings and corresponding implicit answers for model to choose. In this way, once the model find out the correct path, it also gets the correct corresponding answers.

When we get the first hop for a question, there are two possible cases for the second path. One path form is like the upper part in Figure 5, and the other form is like the lower part in Figure 5. For the first one, we start from the first hop's answer to find the next path. In the second situation, we need another path to constrain the answers. Actually, if multiple entities are not identified

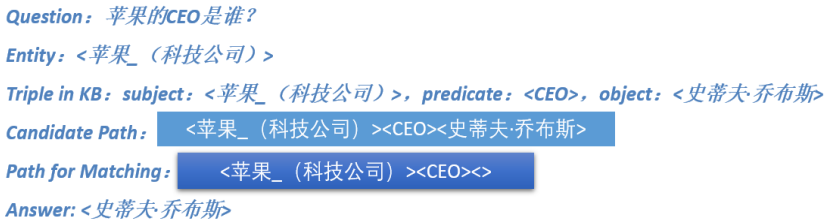

Fig. 4. An example of Matching.

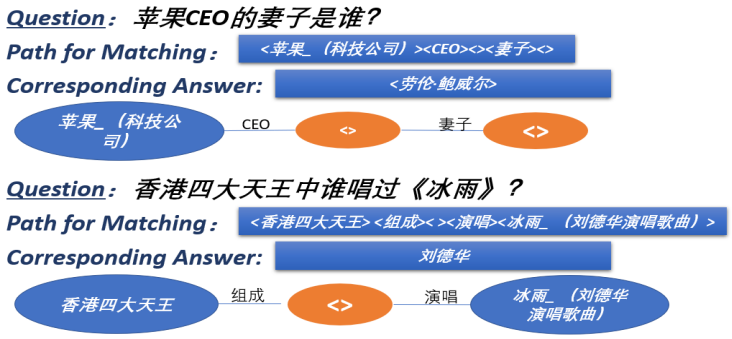

Fig. 5. There are 2 possible cases for 2-hop path. We dig out answer from triple path, and use the left path string for matching. In this way, once the model find out the correct path, it also gets the correct corresponding answers.

in the previous procedures, we can also find the answer for the second case by digging out the entities in different positions.

3 Experiments and Discussion

We evaluate our approach by using CCKS-KBQA-2020 data.The dataset is published by CCKS 2020 evaluation task which includes a knowledge base, knowledge entity mention file, and question-answer pairs for training, validation and testing. The knowledge base has 66 million triples. The knowledge covering common knowledge, finance, health and CONV-19 fields.

We utilize AllenNLP [2] to implement our model. AllenNLP provides high-level API for user to set up NLP model quickly and easily. It can specifies an entire model by a JSON-style configuration file, which facilitates model reuse and distribution.

For Entity Linking , we use the features [5] of {entity's relation count,word overlap of *Question* and *Entity*} and model performance is improved by 1% on F1. For Path Ranking, we use the feature of word overlap between $PathforMatching$ and *Question*, and model performance is improved by 3% on F1. At the stage of prediction, if the model predicts multiple entities, each entity's 1-hop paths

Table 1. Semantic Matching Results for tasks.

Dev. Dataset	Precision	Recall	Top3
Topic Mention Recognition	0.93	0.94	0.95
Entity Linking	0.96	0.97	0.99
First-hop Path Ranking	0.87	0.91	0.93
Multi-hop Path Ranking	0.82	0.85	—

would be sent to First-hop Path Ranking. The model predicts top 3 1-hop paths, and the multi-hop path based on these 1-hop path will be sent to Multi-hop Path Ranking. In the Multi-hop Path Ranking, the model can distinguish the correct path and hop count. Our approach achieves the F1-score of 54.57% on the test set.

4 Conclusions

In this paper, a KBQA system which uses the same model structure to solve all the sub-tasks is presented. The result shows that the proposed system is effective and easy to be reused as a baseline model for future work.

References

1. https://github.com/BettyHcZhang/KBQA_AllenNLP.git
2. Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N.F., Peters, M., Schmitz, M., Zettlemoyer, L.S.: Allennlp: A deep semantic natural language processing platform (2017)
3. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019), http://arxiv.org/abs/1907.11692
4. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 4144–4150 (2017). https://doi.org/10.24963/ijcai.2017/579, https://doi.org/10.24963/ijcai.2017/579
5. Yang Li, Qingliang Miao, C.Y.C.H.W.M.C.H.F.X.: A joint model of entity linking and predicate recognition for knowledge base question answering. CEUR pp. 95–100 (2018)