

SPAN: Understanding a Question with Its Support Answers

Liang Pang*, Yanyan Lan[†], Jiafeng Guo[†], Jun Xu[†], and Xueqi Cheng[†]

CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

*pangliang@software.ict.ac.cn, [†]{lanyanyan, guojiafeng, junxu, cxq}@ict.ac.cn

Abstract

Matching a question to its best answer is a common task in community question answering. In this paper, we focus on the non-factoid questions and aim to pick out the best answer from its candidate answers. Most of the existing deep models directly measure the similarity between question and answer by their individual sentence embeddings. In order to tackle the problem of the information lack in question's descriptions and the lexical gap between questions and answers, we propose a novel deep architecture namely SPAN in this paper. Specifically we introduce support answers to help understand the question, which are defined as the best answers of those similar questions to the original one. Then we can obtain two kinds of similarities, one is between question and the candidate answer, and the other one is between support answers and the candidate answer. The matching score is finally generated by combining them. Experiments on Yahoo! Answers demonstrate that SPAN can outperform the baseline models.

Introduction

Community question answering (CQA) sites have become very popular in recent years (Surdeanu, Ciaramita, and Zaragoza 2008). Information seekers post their questions on the CQA website and other users can give some replies to the question. Therefore, it is valuable if we can automatically select the best answer from the candidate answers.

Recently, deep learning methods have been applied to this task and gain state-of-the-art performances. Most existing deep models (Yu et al. 2014; Qiu and Huang 2015) are directly using similarities between question and answer by their individual sentence embeddings, obtained by convolutional sentence model (CSM). Such deep models are effective in solving the mismatching problem, therefore they usually work well in distinguishing the best answers of one question with those of other questions. However, the information is usually limited in the descriptions of question, and there is always some lexical gap between question and answer in the application of CQA. These issues make the above deep learning approach far from solving the problem of selecting the best answer from the candidates with respect to one question.

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper, we propose a novel deep architecture, namely SPAN, to tackle the above challenges. The main idea comes from the assumption that similar questions usually have similar answers. Based on that, we can better understand a question with its similar questions' best answers, defined as support answers. We can see that support answers can be viewed as a way to provide additional content for a question. Specifically, a deep model is firstly used to generate the sentence embeddings of the question, candidate answer, and the support answers. Then two similarities are computed, one is between question and the candidate answer, and the other one is between support answers and the candidate answer. Finally, the matching score is produced by combining the two similarities. Please note that SPAN is a general architecture where any kind of deep model, such as RNN and LSTM, can be used as the basic component to generate the sentence embedding. In this paper, we use CSM (Kalchbrenner, Grefenstette, and Blunsom 2014) as an example to facilitate the study, mainly because CSM is a common deep model to represent a sentence, and has been widely used in many related works.

Experiments conducted on 100,000 real-world questions from Yahoo! show that SPAN¹ performs better than baseline methods. Besides, we observe that SPAN can also beat baseline methods if we only use support answers themselves in representing a question, indicating that support answers are good representations for a question in the task of CQA.

SPAN: A New Deep Architecture for CQA

In this section, we introduce our new deep architecture SPAN. The basic component of SPAN is convolutional sentence model (CSM), as illustrated in Fig 1A. The input of CSM is a sentence T , where each word w_i in T is represented as its word embedding initialed by Word2Vec². Then one dimensional convolution and pooling are applied layer by layer, and a sentence embedding will be generated.

The architecture of SPAN is illustrated in Fig 1B. Suppose there is a question Q and a candidate answer A , they are both feeded into a CSM to obtain their sentence embeddings, denoted as v_Q and v_A , respectively. Simultaneously, we also

¹More complementary materials of this research are available in <http://pl8787.github.io/qa.html>.

²<https://code.google.com/p/word2vec/>

相似的问题
通常有相似的答案。
support answer
可以为 question
提供额外的信息。
模型的
基本思路

仅使用
support answer
作为问题的
表征也表现
很好, 说明
support answer
的确很有用。

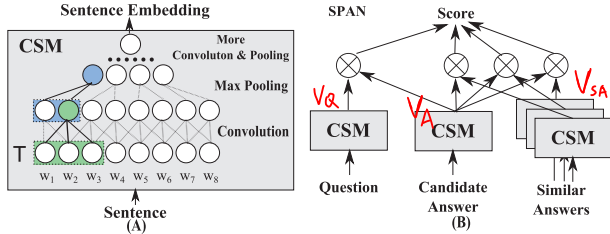


Figure 1: (A) CSM Model, (B) SPAN Model.

leverage the support answers to help understand the semantics of the question. Specifically, we use BM25 (Robertson and Zaragoza 2009), a common retrieval model, to obtain the similar training questions of the original question. Then their best answers are extracted as the support answers, denoted as SA_Q . They are also feeded into CSM to obtain the sentence embeddings $v_{SA}^{(i)}$. Based on those above sentence embeddings, we can obtain two kinds of similarities. The first one is between the question and candidate answer, and the second one is between the support answers and the candidate answer. The similarity measure can be any kind of operators, such as Cosine, Bilinear, and Tensor, denoted as \otimes . The matching score is finally produced by combing these similarities, described as follows.

$$S(Q, A) = \lambda_1 v_Q \otimes v_A + \sum_{i=1}^m \lambda_{2i} v_A \otimes v_{SA}^{(i)},$$

where λ_1 and λ_{2i} are combining parameters, which are tuned by hand on validation set.

All other parameters, such as word embeddings and weights in convolution, are learned in the training process. Specifically, we use the ranking loss for optimization. Given a training question Q and its candidate answers set \mathbb{A} , we denote $B_Q \in \mathbb{A}$ as the best answer of Q . Then we can construct pairs (Q, B_Q, A_i) , where $A_i \in \mathbb{A}, A_i \neq B_Q$. The loss function on each pair is defined as:

$$L(Q, B_Q, A_i) = \max(0, 1 - S(Q, B_Q) + S(Q, A_i)).$$

Experiments

We conduct experiments on Yahoo! Answers dataset³ to evaluate SPAN. The data set contains 142,627 questions and their candidate answers. We first filter out the questions which only contain one candidate answer or have less than three similar questions. The remaining 123,032 questions are splitted into training, validation, and testing set, which contains 98,426, 12,303, and 12,303 questions, respectively.

In testing set \mathbb{Q} , a ranking list of the candidate answers are obtained according to the descending order of the matching scores. The evaluation metrics of our experiments are P@1 and MRR. Since each question only has one best answer, the two evaluation measures are of the following forms.

$$P@1 = \frac{1}{|\mathbb{Q}|} \sum_{Q \in \mathbb{Q}} \mathbb{I}(r_Q = 1), \quad MRR = \frac{1}{|\mathbb{Q}|} \sum_{Q \in \mathbb{Q}} \frac{1}{r_Q},$$

where r_Q denotes the rank of the best answer.

³<http://webscope.sandbox.yahoo.com>

Table 1: Results on Yahoo! Answers (P@1 / MRR).

Model	Random	BM25	CSM	SPAN	SPAN-SA
P@1	25.1	39.4	47.6	48.5	48.3
MRR	48.4	59.9	66.6	67.2	67.1

Random indicate the model of directly selecting a random ranking list for evaluation. BM25 indicate the model of using BM25 to calculate the similarity between question and its candidate answers to obtain the ranking list. The parameters of BM25 are set to be $k_1=0.3$ and $b=0.05$, which are tuned by grid search on validation set. For SPAN, λ_1 and λ_{2i} are set to be equal in our experiments, with other parameters learned automatically. The experimental results are listed in Table 1. From the results, we can see that SPAN outperforms the three baselines. This demonstrates that support answers are good supplements in representing a question, and can effectively facilitate the matching process. We also list the results of SPAN by only using the representations of support answers, denoted as SPAN-SA. The results show that it can still beat the three baselines, indicating that support answers themselves can be viewed as good representations for a question in this question answering task.

Conclusions and Future Work

In this paper, we propose a novel deep architecture for CQA, namely SPAN. The main contribution is to introduce support answers to help understand the semantics of a question. Experimental results show that SPAN performs better than several existing baselines. In future, we decide to test our idea on more complex models such as RNN and LSTM. We also want to investigate how to design an end-to-end model to automatically involve support answers in the learning process.

Acknowledgments

This work was funded by 973 Program of China under Grants No. 2014CB340401 and 2012CB316303, 863 Program of China under Grants No. 2014AA015204, the National Natural Science Foundation of China (NSFC) under Grants No. 61472401, 61433014, 61425016, 61425016, and 61203298, Key Research Program of the Chinese Academy of Sciences under Grant No. KGZD-EW-T03-2, and Youth Innovation Promotion Association CAS under Grants No. 20144310.

References

- Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. *CoRR* abs/1404.2188.
- Qiu, X., and Huang, X. 2015. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*, 1305–1311.
- Robertson, S., and Zaragoza, H. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.
- Surdeanu, M.; Ciaramita, M.; and Zaragoza, H. 2008. Learning to rank answers on large online qa collections. In *ACL*, 719–727.
- Yu, L.; Hermann, K. M.; Blunsom, P.; and Pulman, S. 2014. Deep learning for answer sentence selection. *arXiv:1412.1632*.