A Hybrid Approach to DBQA

Fangying Wu, Muyun Yang(✉), Tiejun Zhao, Zhongyuan Han,
Dequan Zheng, and Shanshan Zhao

Harbin Institute of Technology, Harbin, China
yangmuyun@hit.edu.cn

Abstract. Document-based question answering (DBQA) is a sub-task of open-domain question answering, targeted at selecting the answer sentence(s) from the given documents for a question. In this paper, we propose a hybrid approach to select answer sentences, combining existing models via the rank SVM model. Specifically, we capture the inter-relationship between the question and answer sentences from three aspects: surface string similarity, deep semantic similarity and relevance based on information retrieval models. Our experiments show that an improved retrieval model out-performs other methods, including the deep learning models. And, applying a rank SVM model to combine all these features, we achieve 0.8120 in mean reciprocal rank (MRR) and 0.8111 in mean average precision (MAP) in the opening test.

Keywords: QA · String similarity · Information retrieval · Deep learning · Rank SVM · Hybrid approach

1 Introduction

Document-based question answering (DBQA) is a sub-task of the open-domain question answering. For each question, the target is to select sentences as answers from given relevant document(s). Classic open-domain QA system usually involved three parts: (i) question analysis; (ii) relevant document retrieval; (iii) answer sentence extraction. DBQA is different from open-domain QA in that it just focuses on selecting answer sentences from candidate answer sentences in provided documents.

Previous works on answer selection of QA are dependent on linguist analysis tools and various external resources (Ravichandran 2002). Analyzing sentence structure by such techniques as name entity taggers and shallow parser (Srihari and Li 2000). External knowledge covers semantic resources such as WordNet, QA typology (Hovy et al. 2001), and Wiki pages. A clear limitation of them is that the results are substantially affected by the quality of external resources and parse tools.

Recently, different architectures of deep learning (DL) models are emerging as a new solution to QA task. DL models are either claimed to extract semantic information from texts, or directly adopted to rank candidates by training the distributed representation of question and candidates to find their semantic relevance (Wang and Nyberg 2015; Yin et al. 2015; Severyn and Moschitti 2015).

In another aspect, QA can be naturally regarded as the relevance estimation between question and candidates. For relevance measure, information retrieval circle provides abundant successful solutions. Although IR models have already been

C.-Y. Lin et al. (Eds.): NLPCC-ICCPOL 2016, LNAI 10102, pp. 926–933, 2016.
DOI: 10.1007/978-3-319-50496-4_87

adopted to retrieve relevant documents in QA, they are not well examined for sentence level answer identification yet. Simply treating question as query, candidates as collection, various IR models could be directly applied for DBQA.

Therefore, as an effort in the DBQA evaluation campaign organized by NLP&CC 2016, we focus on examining existing approaches for their efficiency, and then try to combine them for an optimized result. To get a quick and robust method for DBQA, we capture the inter relationship between question and answer sentences from three aspects: surface string similarity, semantic relevance based on DL models and relevance based on IR models. Our experiment results show that features based on IR models perform better than DL models. By applying a rank SVM model to combine all these features, the test results in opening test data set are 0.8120 in mean reciprocal rank (MRR) and 0.8111 in mean average precision (MAP).

The remainder of this paper is organized as follows: Sect. 2 describes the related work. Section 3 describes the detail of different features in our model. In Sect. 4, we present our experiment results, and finally, we draw a conclusion in Sect. 5.

2 Related Work

Many methods have been applied to find the relationship between sentence pairs. Ranking candidate sentences according to bag-of-word matching have been regarded as baseline in many works (Tan et al. 2015; Wang and Nyberg, 2015). But they have a relatively low performance on answer selection. Tran et al. (2015) combine multiple features to rank answers and get the best result in SemEval-2015 Task 3, in which word match is still indispensable. These works show that surface word matching features are necessary for answers selection.

As for the recent efforts to apply DL methods on QA task, the straight-forward idea is to learn the distributed representation of question and answer sentences, then calculate the (cosine) similarity between two sentence vectors (Ming Tan et al. 2015). The more ambitious effort is to train a specialized DL model for similarity estimation. But the best results have been, so far, reported as combination with other features like keywords matches by a machine learning method (Wang and Nyberg 2015; Yin et al. 2015; Severyn and Moschitti 2015).

As for the IR circle, language model is the state-of-the-art model. Compared with learning to rank, the best performed machine learning techniques in web search experiments, language model are still deemed enough to prove a new feature or new strategy designed for the retrieval process. Therefore, in this paper, we do not exhaust variants of learning to rank methods in search, but simply choose two classic forms of language model, the query likelihood model and the KL divergence model.

3 Hybrid Approach via Rank SVM

In this paper, our work focuses on mining the inter relationship between question and candidate answer sentences from three perspectives: surface string similarity, sentence relevance based on retrieval model and relations based on DL text distribution representation. In order to find a quick and robust answer selection model which doesn't depend on extra resources to train, we use linear rank SVM model to combine all features of the three types for ranking candidate answer sentences.

3.1 Measures for Surface String Similarity

Various methods have been used to compare the matching between two sentences (Turney 2006). Especially, the recent success of automatic machine translation evaluation circle also provides another group of string similarity metrics based on n-gram matching (Kondrak 2005), e.g, BLEU, ROUGE and NIST (Papineni et al. 2002; Lin et al. 2004). In our work, we collect the following measures to capture surface string matching between question and the answer sentences:

- Recall and precision of n-gram: Recall and precision of n-gram matches for question and candidate answer sentences.
- MT evaluation metrics: BLEU-1, BLEU-2, BLEU-3, BLEU-4 and NIST5.
- Longest Common Sub-sequence(LCS): The sum of all the same sub-sequence length between question and candidate sentences.
- Edit distance: Defined as the least steps to transform the question sentence to candidate sentence using insertion, deletion and substitution.
- Tf-isf sum: For each matched n-gram, we multiply its frequency by its "reverse sentence frequency" to get a new weight. Then add all the value for each n-gram.
- Words match: The number of same words in question and candidate sentences.
- Nouns match: The number of same nouns in question and candidate sentences.
- Verbs match: The number of same verbs in question and candidate sentences.
- Cosine similarity: Question and candidate sentences are assumed to be two vectors in a |V|-dimensional vector space. |V| is the number of unique terms in the corpus. Calculate the cosine of the angle between two vectors.

In addition, the organizer of the evaluation provided another kind of metric: the translation probability of each candidate sentence given the question. We add it to our system to measure the similarity of question and answer sentences.

3.2 Features Based on Retrieval Models

Here we apply IR-based techniques (Lavrenko et al. 2001; Manning et al. 2008; Huerta 2010) to estimate the relevance between two sentences.

Query Likelihood Model. Given a query sentence Q, we rank the candidate answer sentences according to probability $P(C_i|Q)$. According to Bayes' Rule, we can calculate this by:

$$P(C_i|Q) = \frac{P(C_i)P(Q|\theta_{C_i})}{P(Q)} \tag{1}$$

where θ_{C_i} is the distribution of candidate sentence C_i. We set the prior probability of each candidate sentence C_i equally, so $P(C_i|Q) \propto P(Q|\theta_{C_i})$. $P(Q|\theta_{C_i})$ is the probability that the query text could be generated by the candidate sentence language model and calculated by:

$$P(Q|\theta_{C_i}) = \prod_{j=1}^{n} P(q_j|\theta_{C_i}) \tag{2}$$

where q_j is a query word and question sentence Q contains n query words. We need estimate the language model for each candidate sentence.

According to the maximum likelihood estimator, we estimate the sentence language model $P(w|\theta_{C_i})$ by $\frac{f_{w,C_i}}{|C_i|}$. f_{w,C_i} is the number of times word w occurs in candidate sentence C_i. $|C_i|$ is the total number of words in C_i.

For a word w who doesn't occur in candidate C_i, we apply Dirichlet smoothing method to mitigate the zero-probability problem:

$$P(w|\theta_{C_i}) = (1-\lambda)\frac{f_{w,C_i} + \mu P_{ml}(w|C)}{|C_i| + \mu} + \lambda P_{ml}(w|C) \tag{3}$$

By analyzing all the queries, question words like what and who occur frequently and they put forward questions. Other words in a question sentence are called query words. The query words nearer to a question word are more important. So we re-estimate the sentence language model by:

$$P'(q_j|\theta_{C_i}) = \frac{1}{Z} * \lambda * e^{-\lambda|pos(q_j) - pos(q_c)|} \tag{4}$$

where $pos(q_j) - pos(q_c)$ is the distance between word q_j and word q_c. Z is the regularization term which is the sum of $\lambda * e^{-\lambda|pos(q_j) - pos(q_c)|}$ for all q_j.

3.3 Features Based on Deep Learning

Word Embedding. One of the baseline systems provided by the task trains a word embedding model (Mikolov et al. 2013). It represents a sentence as vector by computing the average of word vectors that occur in this sentence. The cosine of candidate and query sentence vectors is regarded as the probability that the candidate would be an answer.

BLSTM Model. In this paper, we use BLSTM architecture for answer selection (Wang and Nyberg 2015) because it can learn from context information. Figure 1, without attention module within the dashed box, shows the main structure of this model.

Putting embeddings transformed from words into the LSTM sequentially, then we get a matrix that represents semantic of the question or answer sentence. After max-pooling, a vector are generated, which represents the semantic of a sentence. finally, we put the two sentence vectors into a softmax layer to compute the relation of the two sentences. Output of the architecture is a probability that presents the possible of candidate sentence being an answer. We train our model to maximize the probability of answer sentence.

Attention based BLSTM Model Attention model is always used to make the predicate representation matched with predicate-focused sentence representation more effectively (Yin et al. 2015; Zhou et al. 2016; Severyn 2015). Answer sentences use specific words to answer specific question such as name of people to who, words that represent time to when. To better utilize the relationship among words in question and answer sentence, we multiply hidden units by an alignment matrix to generate attention and add it to input question's BLSTM units. Figure 1 as a whole shows the structure of this model.

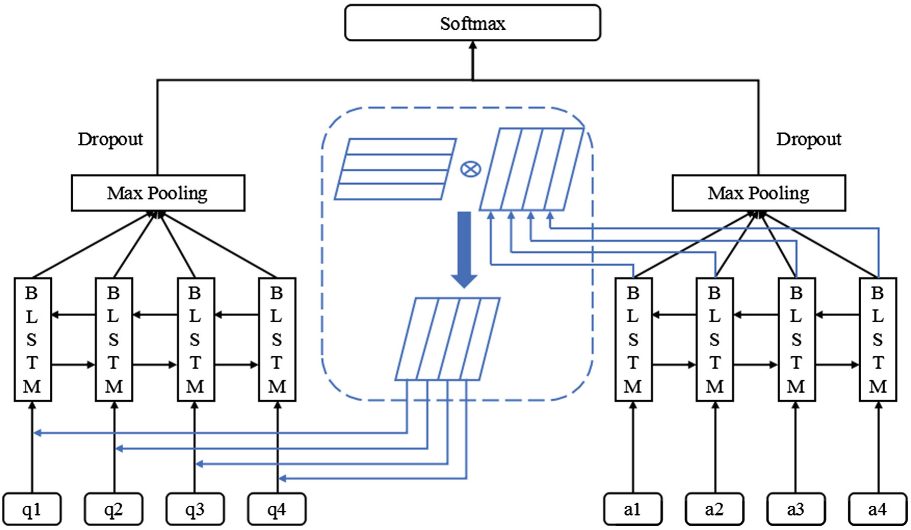

Fig. 1. BLSTM model (with attention)

4 Experiments

4.1 Evaluation Metrics

The evaluation metrics of DBQA system are mean reciprocal rank (MRR) and mean average precision (MAP). MRR and MAP are defined as formulas (5) and (6):

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{5}$$

$$MAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|}AveP(C_i, A_i) \tag{6}$$

where $|Q|$ is the total number of questions in evaluation set and $rank_i$ is the position of the first answer sentence in your sorted candidate answer sentences set, and $AveP(C_i, A_i) = \frac{\sum_{k=1}^{n}(P(k)\cdot rel(k))}{\min(m,n)}$ denotes the average precision.

4.2 Experiments Results and Analysis

The provided training data set contains 8772 questions and a document for each question. We perform 4-fold cross validation to examine the hybrid approach. And we also set 4 different conditions with different feature combination:

- Group 1: features based on surface word matching.
- Group 2: features in group 1 plus features based on query likelihood models.
- Group 3: features in group 1 plus features based on deep learning models.
- Group 4: all features we proposed.

Evaluation results of four group features are showed in Table 1. And contribution of each typical single feature for three aspects is in Table 2. It reveals that the most effective features are that based on query likelihood models and BLSTM model. MRR of these features reach up to 0.65 while others are almost not larger than 0.5.

Table 1. Evaluation of different feature sets

Feature set	MRR	MAP
Group 1	0.6151	0.6126
Group 2	0.7320	0.7292
Group 3	0.6870	0.6848
Group 4	0.7854	0.7822

Features based on string similarity don't make a considerable effect individually. Putting them together, we have gained 0.1 improvement for both MRR and MAP evaluate metrics. It generates a remarkable result after adding features based on query likelihood models to group 2. The value of MRR is 0.7320 and of MAP is 0.7292.

Features based on DL models are somewhat inferior to the IR model. And for DL models, BLSTM model with attention has lower evaluation results compared with that without attention. This situation is strange as related work having showed an opposite result. We think this may arise from the very limited training data provided for the deep training, which usually depends on a very large corpus.

Table 2. Typical single feature of three aspects

Feature	MRR	MAP	Feature	MRR	MAP
2-gram match recall	0.5259	0.5239	Bleu-2	0.4974	0.4961
2-gram match precision	0.4763	0.4755	Bleu-4	0.4221	0.4208
TFISF sum for 2 g	0.4531	0.4517	NIST5	0.4995	0.4981
Word match recall	0.5193	0.5173	LCS	0.5079	0.5058
Cosine similarity	0.4502	0.4493	Edit distance	0.2098	0.2097
Nouns match	0.4929	0.4905	Verbs match	0.4317	0.4293
Translation probability	0.2617	0.2615	Word embedding	0.4769	0.4754
Query likelihood model	0.6536	0.6514	BLSTM	0.6634	0.6615
Re-estimate query likelihood model	0.6936	0.6907	BLSTM with attention	0.5743	0.5720

For candidate answer sentences among which models of feature group four and five give a highest score to the real answer, we analyze their corresponding questions. Query likelihood models perform differently from BLSTM models. Generally speaking, language model performs better than DL models in training data set. But when question sentences contain specific question words like how many, DL models perform better than language model.

Finally, we combine all these features of three types. The results of four-fold cross validation are 0.7854 and 0.7822 for MRR and MAP, respectively. That is, all the features are utilized and training over the whole provided data to be tested by the final open text, achieving a result of 0.8120 in MRR and 0.8111 in MAP.

5 Conclusion

In this paper, we examine the existing question-answer sentence methods from three perspectives, and propose a hybrid solution by ranking SVM. We reveal that the surface word similarity doesn't work well in selecting answer sentences. In contrast, an improved IR model performs best in this task, better than the state-of-the-art deep learning techniques. Combing all these model results under the rank SVM framework, we achieve in the open test with 0.8120 in MRR and 0.8111 in MAP.

Acknowledgments. This paper is supported by the project of Natural Science Foundation of China (Grant Nos. 61272384, 61402134, and 61370170).

References

Srihari, R.K., Li, W.: A question answering system supported by information extraction. In: Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL-2000), Seattle, WA, pp. 166–172 (2000)

Hovy, E., Hermjakob, U., Lin, C., et al.: The use of external knowledge of factoid QA. In: TREC, vol. 2001, pp. 644–652 (2001)

Ravichandran, D., Hovy, E.: Learning surface text patterns for a question answering system. In: Meeting of the Association for Computational Linguistics (2002)

Severyn, A., Moschitti, A.: Learning to rank short text pairs with convolutional deep neural networks. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (2015)

Tran, Q.H., Tran, V., Vu, T., et al.: JAIST: combining multiple features for answer selection in community question answering. In: Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval, vol. 15, pp. 215–219 (2015)

Tan, M., Santos, C.N., Zhou, B., et al.: LSTM-based deep learning models for non-factoid answer selection. arXiv preprint arXiv:1511.04108 (2015)

Wang, D.W., Nyberg, E.: A long short-term memory model for answer sentence selection in question answering. In: Meeting of the Association for Computational Linguistics (2015)

Yin, W., Yu, M., Zhou, B., et al.: Simple question answering by attentive convolutional neural network. arXiv preprint arXiv:1606.03391 (2016)

Papineni, K., Roukos, S., Ward, T., et al.: BLEU: a method for automatic evaluation of machine translation. In: Meeting of the Association for Computational Linguistics (2002)

Kondrak, G.: *N*-gram similarity and distance. In: Consens, Mariano, Navarro, Gonzalo (eds.) SPIRE 2005. LNCS, vol. 3772, pp. 115–126. Springer, Heidelberg (2005). doi:10.1007/11575832_13

Lin, C., Och, F.J.: Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: Meeting of the Association for Computational Linguistics (2004)

Huerta, J.M.: An information-retrieval approach to language modeling: applications to social data. In: North American Chapter of the Association for Computational Linguistics (2010)

Manning, C.D., Raghavan, P., Schutze, H., et al.: Introduction to information retrieval. In: Proceedings of the International Communication of Association for Computing Machinery Conference (2008)

Lavrenko, V., Croft, W.B.: Relevance based language models. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (2001)

Turney, P.D.: Similarity of semantic relations. J. Comput. Linguist. **32**(3), 379–416 (2006)

Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. In: Neural Information Processing Systems (2013)

Yin, W., Schütze, H., Xiang, B., et al.: ABCNN: attention-based convolutional neural network for modeling sentence pairs. arXiv preprint arXiv:1512.05193 (2015)