ICL00 at SemEval-2016 Task 3: Translation-Based Method for CQA System

Minghua Zhang Yunfang Wu
Institute of Computational Linguistics, Peking University

Abstract

We participate in the English subtask B and C at SemEval-2016 Task 3 "Community Question Answering". This paper is concerned with the description of our participating system. We propose a ranking model that combines a translation model with the cosine similarity-based method. Compared to the traditional bag of words method, the proposed model is more effective because the relationships between words can be explicitly modeled through word-to-word translation probabilities. Experiments conducted on the official test data demonstrate that our proposed ranking method obtains promising results.

1 Introduction

The SemEval-2016 Task 3 (Nakov et al., 2016) Community Question Answering (CQA) covers a full task on CQA and which is, therefore, closer to a real application. To facilitate the participation of non IR/QA scholar to the task, the search engine step is already carried out which means that the task organizers explicitly provides the set of potential answers to be reranked. That is to say, given a new question (aka original question) and the set of the first 10 related questions (retrieved by a search engine) in subtask B, our system will focus on reranking the related questions according to their similarity with the original question. Similar to subtask B, in subtask C that is the main English subtask, Given a new question and the set of the first 10 related questions, each associated with its first 10 comments appearing in its thread, we will rerank the 100 comments (10 questions * 10 comments) according to their relevance with respect to the original question.

Note that the subtask B will give us enough tools to solve the main subtask. And therefore, our paper will give an overall description of the system based on subtask B. In section 2, we will briefly discuss an important difference between the subtask C and subtask B in the course of reranking.

However, the major challenge for subtask B, as for most of CQA systems, is the word mismatch between the original question and the related question. For example, "Where I can buy good oil for massage?" and "Is there any place I can find scented massage oils in Qatar?" are two very similar questions, but they only have a few words in common. To solve the word mismatch problem, we focus on translation-based approaches in this paper.

The remainder of this paper is organized as follows. Section 2 introduces the methods used in the ranking model clearly. Section 3 presents the translation probabilities Estimation. We will talk about an overview of the system in section 4. Section 5 presents the experimental results. In Section 6, we conclude with ideas for future research.

2 Ranking Approach

In SemEval-2016 Task 3, the dataset file is a sequence of question pair instances. Each instance contain an original question and a thread which consists of a potentially related question, together with 10 comments for it. Next, let Q-Q denotes the set of all original-related question pairs in the archive, Q-Q = {…, [(org_i, rel_1), (org_i, rel_2), ..., (org_i, rel_j), …, (org_i, rel_{10})], …}. So, the goal of subtask B is to rerank the related questions according to the Score(org_i, rel_j). Typically, this score can be modeled by the probability that org_i is generated by rel_j. Thus, the following part of this section focus on how to calculate $P(org_i|rel_j)$.

Proceedings of SemEval-2016, pages 857–860,
San Diego, California, June 16-17, 2016.

However, the job of subtask C is to rerank the 100 comments according to their relevance with respect to each original question. For clarity, let Q-C denotes the set of all question-comment pairs, Q-C = {…, [(org_i, rel_j, C_{ij1}), (org_i, rel_j, C_{ij2}), ..., (org_i, rel_j, C_{ijk}), …, (org_i, rel_j, C_{ij100})], …}. To begin the process, we apply the tool which is designed for subtask B to calculate the relevance between original question and related question, then regard it as a weight. In the next step, we make use of translation model to obtain the relevance between original question and comment. Finally, we will rerank the comments according to the Score(org_i, C_{ijk}) which can be written as:

$$Weight_j = \frac{P(org_i|rel_j)}{\sum_{j'=1}^{n} P(org_i|rel_{j'})} \quad (1)$$

$$Score(org_i, C_{ijk}) = Weight_j * P_{trans}(org_i|C_{ijk}) \quad (2)$$

where org_i is the original question, rel_j is the related question, and C_{ijk} is the comment for rel_j.

2.1 Word-Based Translation Model

Previous work (Jeon et al., 2005) were the first to apply the translation based method to CQA, subsequent work (Xue et al., 2008) proposed to linearly combine language model and word-based translation model into a unified framework. The experiments show that this model gains better performance than both the language model and the word-based translation model. Following Xue et al. (2008), this model can be written as:

$$Score(org_i, rel_j) = \prod_{w \in org_i} P(w|rel_j) \quad (3)$$

$$P(w|rel_j) = \alpha \frac{\#(w, rel_j)}{|rel_j|} + \beta \sum_{t \in rel_j} P(w|t) \frac{\#(t, rel_j)}{|rel_j|} + \gamma \frac{\#(w, B)}{|B|} \quad (4)$$

where #(w, rel_j) and #(t, rel_j) is the frequency of term w and t in rel_j respectively, B denotes the whole archive, $|rel_j|$ and |B| denote the length of rel_j and B respectively, P(w|t) denotes the translation probability from word t to word w.

2.2 Combination of Cosine similarity and Translation Method

We compared the performances of a unigram language model and a cosine similarity-based method on the development dataset, which demonstrated that the cosine similarity method outperforms the language model. Therefore, we propose to linearly combine the cosine similarity and translation model into a ranking model, which can be written as:

$$Score(org_i, rel_j) = \alpha P_{cos}(org_i, rel_j) + \beta P'_{trans}(org_i|rel_j) \quad (5)$$

$$P_{trans}(org_i|rel_j) = \prod_{w \in org_i} (\sum_{t \in rel_j} P(w|t) \frac{\#(t, rel_j)}{|rel_j|}) \quad (6)$$

$$P'_{trans}(org_i|rel_j) = {}^{10}/_{-\log_2 P_{trans}(org_i|rel_j)} \quad (7)$$

where $P_{\cos}(org_i, rel_j)$ denotes the cosine similarity. $P_{\text{trans}}(org_i|rel_j)$ denotes the translation probabilities. The two parts are not in an order of magnitude. So when we obtained the two similarity scores, the translation probabilities have to be transformed according to the formula (7).

3 Translation Probabilities Estimation

3.1 Parallel Corpus Collection

The performance of the proposed ranking model heavily depends on the quality of the translation probabilities. Therefore, besides designing translation based ranking method, another important problem is how to learn good word-to-word translation probabilities.

In the given training dataset, the similarity relationship between the related question and original question can be accessed from the attribute "RELQ_RELEVANCE2ORGQ" belonging to the tag "RelQuestion", which can be PerfectMatch, Relevant and Irrelevant. When the attribute value takes PerfectMatch or Relevant, there's a strong possibility that the original question and relevant question express similar meanings with different words. So, it is natural to use the matching original-relevant question pairs as the "parallel corpus" to estimate word-to-word translation probabilities. Furthermore, it is easy to realize that if one original question is similar with two different relevant questions simultaneously, then the two relevant questions would also express similar meanings. As an example, from the initial parallel corpus {[org_1, rel_1], [org_1, rel_2], [org_1, rel_3], [org_2, rel_4], [org_2, rel_5]}, we can obtain the new big parallel corpus {[org_1, rel_1], [org_1, rel_2], [org_1, rel_3], [rel_1, rel_2], [rel_1, rel_3], [rel_2, rel_3], [org_2,

rel_4], [org_2, rel_5], [rel_4, rel_5]} through the simple extension method.

In the IBM translation model 1, sentences are normally translated from one language into another language. But in our task, the similar sentence pairs are written in the same language, the correspondence of words is not as strong as in the bilingual sentence pair. The word-to-word translation probabilities can be learned with either part as the source language and the other part as the target. Accordingly, the training data is doubled. When there is a parallel corpus consisting of the similar sentence pairs, the training module will utilize IBM translation model 1 incorporating an EM-based algorithm to learn the word-to-word translation probabilities.

3.2 Consolidation Method

The parallel sentence pairs are written in the same language. If source sentences contain word "wi" and target sentences contain word "wj", we can obtain the word-to-word translation probabilities $P(wj|wi)$. Conversely, the word "wi" can also appear in target sentences and the word "wj" appear in source sentences. So, we can obtain the reverse translation probabilities $P(wi|wj)$ through the same training process. Then we assume that the reverse translation probabilities are additional information to improve the word-to-word translation probabilities, and the combination of both will be consolidation beneficial. So we linearly combine the trained word-to-word translation probabilities:

$$P_{lin}(wi|wj) = \gamma P(wi|wj) + (1-\gamma)P(wj|wi) \qquad (8)$$

To see how much the consolidation strategy benefits the rerank task, we introduce two baseline methods for comparison. The first method denotes the initial word-to-word translation probabilities, and the second denote the reverse translation probabilities. Table 1 reports the experimental results of Subtask B on the development dataset. We can see that the consolidation strategy outperforms the two baseline methods in our task. From the experimental result, we can see that the reverse translation probabilities do have some positive effects.

Model	Trans Prob	MAP
$Score(org_i, rel_j)$	$P(wj\|wi)$	0.7312
	$P(wi\|wj)$	0.7376
	$P_{lin}(wi\|wj)$	0.7415

Table 1: The impact of consolidation strategy.

4 System overview

In the following part, we will introduce the details of the implementation. The whole calculation process can be divided into two main modules: Pre-processing and Estimate.

Pre-processing. This module tries to extract the subject text and the main body of questions from the XML-formatted input file first of all, then combine the two parts together to form the original question and related question. Next, the system makes word segmentation for each pairs of original question and related question, and removes stop words at the same time. Furthermore, Porter stemmer is employed to extract stem, which will be beneficial to learn good word-to-word translation probabilities. For the sake of saving the evaluation times, we execute statistical calculations on the word list which represents the original question and related question after segmentation. For example, there is a words list [massage, oil, where, buy, good, oil, massage], we'll obtain a dictionary {massage:2, oil:2, where:1, buy:1, good:1} after statistics, and we also know that the length of list is seven.

Estimate: In this stage, the system loads the word-to-word translation probability table, which we select defaultdict[1] as its storage structure. Finally, we compute the similarity score of original question and related question according to our ranking model which linearly combines the cosine similarity and translation model. In subtask B, the labels PerfectMatch and Relevant should be regarded as equal, so our goal is to rank the PerfectMatch and Relevant candidates at the top, in any order, and the Irrelevant candidates at the bottom, also in any order.

5 Experiments

In this section, experiments are conducted on DEV

[1] Defaultdict is a subclass of the dict that calls a factory function to supply missing values in Python.

dataset to demonstrate the performance of our proposed ranking model.

5.1 Data Set and Evaluation Metrics

The official dataset contains TRAIN, DEV and TEST. The development dataset is intended to be used as a development-time evaluation dataset as we develop our systems. However, when submitting prediction results, we will add the development dataset to training data as well. The total available data of the TRAIN part is made up of 267 original questions and 2669 related questions for Subtask B, plus 26690 related comments for Subtask C. DEV dataset which were manually double-checked and are very reliable consist of 50 original questions, 500 related questions, and 5,000 comments. As far as the TEST is concerned, the task organizers provide participants with 70 original questions, so, there would be 700 predictions for subtask B and 7,000 predictions for subtask C. The official scorer will provide a number of evaluation measures to assess the quality of the output of a system, but the official evaluation measure towards which all systems will be evaluated and ranked is Mean Average Precision (MAP).

5.2 Results

Five types of baselines are used to compare with our proposed ranking model. Table 2 presents the MAP performance comparison of different methods on DEV dataset in subtask B. Row 1 to row 5 are the IR engine default ordering, Language Model, Cosine Similarity-based method, Translation Model and Translation-based Language model. Row 5 is our proposed translation-based reranking method. Compared with other baselines approaches, our proposed model received good results. Finally, the competition result for our primary submissions are 0.7511 against 0.7475 baseline in Subtask B, and 0.4919 against 0.4036 baseline in Subtask C.

6 Conclusions and Future Work

In this paper, we propose a ranking model that combines a translation model with the cosine-based similarity method to solve the rerank task in CQA. Experiments on test data demonstrate the effectiveness of our method.

There are some ways in which this research could be continued. First, we plan to apply neural machine translation (Bahdanau et al., 2014) to learn good translation probabilities. In addition, phrase-based translation model for question retrieval (Zhou et al., 2011) have shown superior performance compared to word-based translation models. So it is necessary to try this method to further improve the performance.

#	Methods	MAP
1	IR-engine	0.7135
2	LM	0.7248
3	Cosine	0.7287
4	Trans	0.7342
5	Trans-LM	0.7360
6	Trans-Cosine	0.7415

Table 2: Comparison of different methods in subtask B.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61371129), National High Technology Research and Development Program of China (2015AA015403), Key Program of Social Science foundation of China (12&ZD227).

References

Bahdanau D, Cho K, Bengio Y. *Neural Machine Translation by Jointly Learning to Align and Translate[J]*. Eprint Arxiv, 2014.

J. Jeon, W. B. Croft, and J. H. Lee. *Finding similar questions in large question and answer archives*. In *Proceedings of the 14th ACM Conference on Information and Knowledge Management*, pages 84–90, 2005.

Preslav Nakov, Llu\'{i}s M\`{a}rquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community Question Answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, CA.

X. Xue, J. Jeon, and W. B. Croft. 2008. *Retrieval models for question and answer archives*. In *Proceedings of SIGIR*, pages 475-482.

Zhou G, Cai L, Zhao J, et al. *Phrase-Based Translation Model for Question Retrieval in Community Question Answer Archives[C]// Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*: Human Language Technologies - Volume 1. Association for Computational Linguistics, 2011:653-662.