Nders at the NTCIR-12 STC Task: Ranking Response Messages with Mixed Similarity for Short Text Conversation

Ge Xu
Minjiang University, Fuzhou, China
xuge@pku.edu.cn

Guifang Lu
NetDragon Websoft Inc.,Fuzhou, China
luguif@nd.com.cn

ABSTRACT

Short Text Conversation (STC) is a typical scenario in man-machine conversation, which simplifies the conversation into one round interaction and makes the related tasks more practical.

This paper presents a simple approach to the Chinese STC task issued by NTCIR-12. Given a repository of post-comment pairs, for any query, we define three types of similarity and merged them according to empirical weights. We consider the similarity between a query and a post/comment. To catch more logic relevance, we train a LDA model to map a query/comment into a topic space, and then calculate the similarity between them.

The evaluation results show that our approach performs better than average. Considering its simplicity, our approach can be used in quickly deploying related STC systems.

Team Name

Nders

Subtask

Short Text Conversation Task (Chinese)

Keywords

Short Text Conversation, Semantic Similarity, Information Retrieval, Topic Model

1. INTRODUCTION

Conversation with computers is a hard NLP task, which involves language understanding, reasoning, and use of common sense knowledge. NTCIR offers a simplified version of the problem: one round of conversation formed by two short texts, with the former being a message from human and the latter being a response to the message from the computer. It is referred as **short text conversation (STC) [1]**. Many customer service robots in developing can be seen as STC, which only considers one round conversation and the messages are normally short.

For the Chinese STC task released by NTCIR, we consider a simple but effective method to solve it. We first propose several types of similarity, and then merge all the similarity scores with weights. Finally, we provide the response comments according to the ranking score.

2. APPROACH

The task can be formulated as following: given a repository of post-comment pairs, for any query, choose the best comment from the repository. The best fetched comment should have the following merits:

1) **Logic Consistency**: Responses should be logically consistent with the test post;
2) **Semantic Relevance**: Responses should be semantically relevant to the test post;
3) **Scenario Dependence**: A specific scenario (context) can be taken into consideration to make a comment appropriate;
4) **Repeating Opinions**: Responses can repeat the same opinions as presented in their posts;

Repeating Opinions and **Semantic Relevance** can be properly processed by using the current NLP techniques. Fortunately, Lots of **Logic Consistency** and **Scenario Dependence** are entangled with semantic relevance. Therefore, for simplicity, we only consider semantic relevance in our experiments.

In this paper, we at first introduce three types of similarity used for our STC Task, and then we give the formula used to mix three similarity scores. Finally, we choose the best response comments from the repository according to mixed similarity scores.

To calculate the similarity between the texts of various lengths, we normally represent the texts by fixed-length vectors, so we can calculate the similarity with these vectors conveniently. We use different vector models in different types of similarity.

2.1 Query-Post Similarity

At first, we measure the similarity between a query and any post fetched from the repository of post-comment pairs. After transverse all the posts, and obtain their similarities with the query, we keep the top 10 posts by similarity, and use their comments as the candidates.

We use the TF-IDF [1] vector space model for measuring the query-post similarity [5], so the query and the post are all translated into vectors with same length, namely q and p.

$$s_{qp}(q,p) = \frac{q^T p}{\|q\|\|p\|} \quad (1)$$

This type of similarity calculates the relatedness of a query and a post, with the assumption that **similar questions tend to have similar comments**. So we can find the similar questions with the given query, and use their comments as potential responses. In our experiments, this type of similarity proved to be more important than other two types, so we assign it higher weight when mixing with others.

2.2 Query-Comment Similarity

According to the task requirements, responses should have semantic relevance with the given query. We use the same method in section 2.1 to calculate the similarity between a query

[1] In calculating *tf* and *idf*, a document is a post or a comment.

and any comment from the repository, which is shown in the following:

$$s_{qc}(q,c) = \frac{q^T c}{\|q\|\|c\|} \quad (2)$$

where q and c are the TF-IDF vector respectively, and the result is the similarity between the query q and comment c.

This similarity can catch part of **Semantic Relevance.**

2.3 Matching with Topic Model

TF-IDF is the commonly used vector model for information retrieval, and can produce good results in general. But it only takes into account the statistical properties of the word in the context without considering the semantic information of the word itself. Many topic models (such as LDA [3], LSI [4] etc.) can help deal with term mismatch in search, and provide the similarity between different words, thus improve the similarity calculation between two texts. In another word, queries and comments that have similar topic distribution are likely to be relevant.

In our experiments, we see each post-comment pair as a document, and then train a LDA model only using data in the repository, then we map queries and comments to the vectors of 500 topics, and calculate the query-comment similarity with the following formula:

$$s_{topic}(q,\mathrm{c}) = \frac{{q_{topic}}^T c_{topic}}{\|q_{topic}\|\|c_{topic}\|} \quad (3)$$

The STC task released by NTCIR requires that the fetched comments should have both semantic and logic relevance with the query. However, in this stage of NLP development, no satisfying tools or approach can offer high-quality logic relevance between two texts. So we only calculate the semantic relationship between the query and the comments. This similarity can catch main **Semantic Relevance** and part of **Logic Consistency** and of **Scenario Dependence.**

2.4 Ranking score

We define the score of a candidate comments as following:

$$score(q,(p,c)) = \sum w_i \Phi_i(q,(p,c)) \quad (4)$$

The score is a weighed sum of the three types of similarity scores, where $\Phi_i(q,(p,c))$ is the score of the i-th similarity between q and p/c , and w_i is the weight of i-th similarity. In our experiments, by experience, we set the weights to 0.5, 0.25 and 0.25 respectively. We find that finding similar posts with a query can provide better performance, so we assign larger weight to *Query-Post Similarity*. We did not use the train dataset to tune the weights due to time limit, although it may provide better weight setting.

For *Query-Post Similarity*, we are calculating the similarity between a query and a post. To pass similarity scores to the comments, we assign the *Query-Post Similarity* score to all the query's comments.

For a given query, we select comments with the top mixed scores from the repository of post-comment pairs.

3. Experiments

3.1 Experimental Setting

A repository of post-comment pairs from Sina Weibo is offered. The repository is about 800MB, contains 196,495 Weibo posts and the corresponding 4,637,926 comments, and finally we obtained 5,648,128 post-comment pairs. So each post has 28 different comments on average, and one comment can be used to respond to multiple different posts.

We use jieba[2] package to perform word segmentation without POS tagging. Approximately, to retrieve 10 best comments for a query takes one minute, the experiment for the submitted run takes about two hours.

In all unaccounted cases, we use the default settings.

3.2 Evaluation Result:

We conduct many experiments with the given post-comment pairs, and submit one run[3] for official evaluation, which we think is the best by our checking. There are totally 16 teams and 44 submitted runs for Chinese STC task [2]. Our run ranks 22nd, 14th and 8th according to evaluation measures of nDCG@1, P+ and nERR@10 respectively.

According to the organizer of this task, nERR@10 is the primary measure of STC, our run ranks 8th in 44 runs and we rank our 4th as a team in 16 teams. By contrast, when using nDCG@1, our run ranks in the middle of all runs. Since our approach didn't consider the **Logic Consistency** and **Scenario Dependence** separately, so the top retrieved response may suffer from insufficiency of these two kinds of information. However, when the top 10 responses are considered, our approach behaves much better in all submitted runs when being evaluated by nERR@10. We guess that **Logic Consistency** and **Scenario Dependence** are still highly correlated with semantic relevance (Tf-Idf, LDA, and Word2Vec etc.), and current technology is not strong enough to separate these kinds of relevance precisely.

4. CONCLUSIONS

The results are a bit better than we expected, since we use a very simple solution to the task, which shows that our approach can be used in quickly deploying a STC system. We hope to communicate with more researchers in the workshop of this task.

5. REFERENCES

[1] Zongcheng Ji, Zhengdong Lu, Hang Li , An Information Retrieval Approach to Short Text Conversation, arXiv preprint arXiv:1408.6988, 2014

[2] LifengShang, Tetsuya Sakai, Zhengdong Lu, Hang LI, Ryuichiro Higashinaka, Yusuke Miyao. Overview of the NTCIR-12 Short Text Conversation Task, NTCIR-12,Techincal Report, 2015

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 3:993–1022, March 2003.

[2] https://pypi.python.org/pypi/jieba

[3] Our team name is Nders

[4] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K.Landauer, and Richard Harshman. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6):391–407, 1990.

[5] Salton, G. and McGill, M. J. 1983 Introduction to modern information retrieval. McGraw-Hill