A Combination of Similarity and Rule-based Method of PolyU for NTCIR-12 STC Task

Chuwei Luo
Department of computing,
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
cscluo@comp.polyu.edu.hk

Wenjie Li
Department of computing,
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
cswjli@comp.polyu.edu.hk

ABSTRACT
In this report, we describe the approach we use in NTCIR-12 Short Text Conversation task. Because we register this task too late and we only have less than one week to do this task, we design a simple approach that is based on cosine similarity of sentence and some handcrafted rules. The result shows the effectiveness of our method.

Team Name
PolyU

Subtasks
Short-text-conversation (Chinese subtask).

Keywords
Short-text-conversation, cosine similarity, handcrafted rules.

1. INTRODUCTION
Natural language conversation between computer and human is a challenging task in natural language processing. Short Text Conversation (STC) is one round of conversation formed by two short texts, with the former being an initial post from a user and the latter being a comment given by the computer. In NTCIR-12 STC task, it's an information retrieval problem. Given a post, we should choose 10 appropriate comments from the large post-comments repository [1,2,3,4].

Our team participated in the STC Chinese subtask. This paper describes approach we use in this task. Because we register to this task late and we only have less than one week to do this work, we design a very simple approach (See Section 2) to do this task. The human evaluation results show the effectiveness of our approach.

2. METHOD
As illustrate in Figure 1, for a given post A, we choose the comment from the comment repository with the highest ranking score. Where the score is calculated as follows:

$$Rank_{score} = \alpha \times socre_1 + \beta \times score_2 + \gamma \times score_3$$

$Score_1$ is the similarity between the test post and the post in post repository. $Score_2$ is the similarity between the candidate posts that we choose by ranking the $Score_1$ and the corresponding comments to the candidate posts. $Score_3$ is the similarity between the test post and the comments that are the real comments to the candidate posts we choose. α, β and γ are the weight of the 3 scores. And after we pick the candidate comments according to the $Rank_{score}$, we design some rules to re-rank the candidate comments. Finally, we pick 10 comments as the final chosen comments.

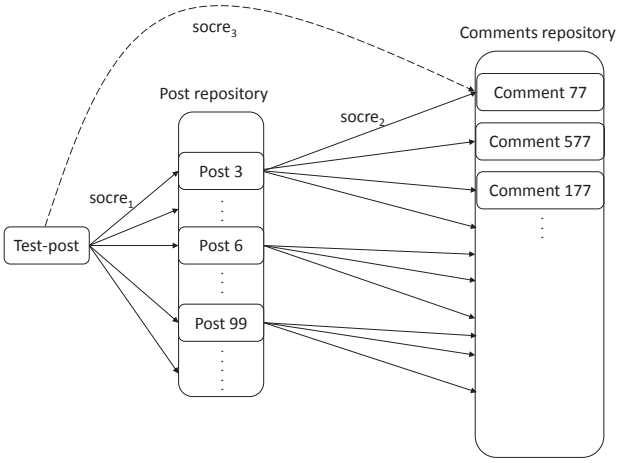

Figure 1 Our Method

Calculate $Score_1$ and choose candidate comments:

For lack of computational ability of our computer and in order to get candidate posts with the highest similarity of the given test post as fast as possible, we use information retrieval platform Lucene to pick N posts that have the highest *Lucene's similarity score* which is based on Vector Space Model (VSM) and TF-IDF as $Score_1$ as a candidate posts set $P = (post_1, post_2, \ldots, post_N)$. We can then easily get the comments that is corresponding to the N posts we choose. Suppose a post in post set, for example, $post_i$ have M_j comments. Finally, we choose these comments as our candidate comments set which have $M=M_1+M_2+\ldots+M_N$ comments in it.

Calculate $Score_2$ and $Score_3$:

Before we calculate $Score_2$ and $Score_3$, we train Chinese word embedding[4] on our STC dataset. We calculate sentence embedding by sum up all words embeddings that in the sentence as follows:

$$Sentence_{emb} = \sum word_{i\,emb}$$

After we get sentence embedding and candidate comments, $Score_2$ is calculated by cosine similarity between each one of the chosen N posts and corresponding comments according to sentences

embeddings. For example, $post_i$ have M_j corresponding comments, for $a = 1,2, \ldots, M_j$:

$$score_2 = cosine\_similarity(post_i, comment_a)$$

$Score_3$ is calculated by cosine similarity between the test post and each one of candidate comments according to sentences embeddings. For example, for $b= 1, 2, \ldots, M$:

$$score_3 = cosine\_similarity(test\_post, comment_b)$$

Calculate $Rank_{score}$:

After we get $Score_1$, $Score_2$ and $Score_3$, we use pairwise learning to rank method to get parameter α, β and γ. So every (test post, comment in candidate) pair can get a $Rank_{score}$. At last, we choose the top 10 comments in candidate set by $Rank_{score}$.

Rules for re-ranking:

1. We build a General Comments Database(GCD) which the comment in it can be treated as an appropriate comment to every post like "哈哈"("haha"), "不错"("not bad"), "很棒"("Pretty good") and so on. When the $Rank_{score}$ is below a threshold, we replace this comment with a comment in the GCD.
2. We observed that when the length of a comment in STC dataset is too long, it may contain a lot of other information that could lead to ambiguity. So when we meet a comment like that, we decide not to choose it as a candidate even if it has a higher $Rank_{score}$ than other comments.
3. After we choose 10 posts from the post repository, we treat all the corresponding comments, 10 posts and test post as a "document" and extract top k keywords from this "document". So when we construct the candidate comments, we also consider the number of the word of a comment in the keywords list. If a comment have fewer words in keywords list, we reduce its parameter β's value.

3. Experiments

3.1 Dataset

The NTCIR-12 STC dataset (Chinese) is a very large dataset which have almost over 5 million post-comment pairs from Weibo. There are also about 6 thousand labeled post-comment pairs available for us to train our parameters. As for test, the number of query posts is 100.

3.2 Implementation Details

We use NLPIR Chinese word segmenter[1] to split the posts and comments into sequences of words. We get almost 570 thousand words. To train our word embeddings, we use Gensim[2] a free Python library to do this work. The dimensions of word embeddings are 300. We use RankLib[3] to train our parameters α, β and γ. The parameter N is set to 10 and k is set to 20.

[1] http://ictclas.nlpir.org/.

[2] https://radimrehurek.com/gensim/

[3] https://people.cs.umass.edu/~vdang/ranklib.html

4. Results and Analysis

The evaluation measures used in NTCIR-12 STC (Chinese subtask) are *nG@1*, *nERR@10* and *P+*. We submit 3 runs: PolyU-C-R1, PolyU-C-R2 and PolyU-C-R3. We use rule 1 in PolyU-C-R1, rule 1 and 2 in PolyU-C-R2, rule 1, 2 and 3 in PolyU-C-R1. Table 2 shows the official results for the NTCIR-12 STC of our approach.

The results show that P+ and nERR@10 of our method are higher than all teams' average in PolyU-C-R1 and PolyU-C-R2. The relatively bad performance of PolyU-C-R3 shows that rule 3 may have a bad influence on our method. Rule 1 and 2 can contribute a lot to our method. This means that the similarity of sentences between post and comments is very important and when we pick comments from the comment repository, long comments are easy to bring bad influence than a shorter one.

Table 1 results for the NTCIR-12 STC (Chinese subtask)

Runs	Mean nDCG@1	Mean P+	Mean nERR@10
Average (all team)	0.2120	0.3475	0.3245
PolyU-C-R1	0.1900	0.3510	0.3314
PolyU-C-R2	0.1867	0.3603	0.3426
PolyU-C-R3	0.1667	0.2968	0.2771

5. ACKNOWLEDGMENTS

Thanks to organizers of NTCIR-12 STC (Chinese subtask) for providing short text conversation dataset on Weibo. This dataset is very good for researchers to do other related researches.

6. REFERENCES

[1] Lifeng Shang, Tetsuya Sakai, Zhengdong Lu, Hang Li, Ryuichiro Higashinaka, Yusuke Miyao. 2016. Overview of the NTCIR-12 Short Text Conversation Task. In *NTCIR-12*.

[2] Hao Wang, Zhengdong Lu, Hang Li and Enhong Chen. 2013. A Dataset for Research on Short-Text Conversation. In *Proceedings of EMNLP,* 935–945.

[3] Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

[4] Lifeng Shang and Zhengdong Lu and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of ACL,* 1577–1586.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Prockeedings of NIPS.*