Nders at NTCIR-13 Short Text Conversation 2 Task

Liansheng Lin[1], Han Ni[2], and Ge Xu[3]

[1] Fuzhou University, Fuzhou, China
linliansheng@nd.com.cn
[2] NedDragon Websoft Inc., Fuzhou, China
ni.han.ms6@foxmail.com
[3] Minjiang University, Fuzhou, China
xuge@pku.edu.cn

Abstract. This paper describes our approaches at NTCIR-13 short text conversation 2 (STC-2) task (Chinese). For a new post, our system firstly retrieves similar posts in the repository and gets their corresponding comments, and then finds the related comments directly from the repository. Moreover, we devises a pattern-idf to rerank the candidates from above. Our best run achieved 0.4780 for mean nG@1, 0.5497 for mean P+, and 0.5882 for mean nERR@10, and respectively ranked 4th, 5th, 5th among 22 teams.

Keywords: Short text conversation, LSI, Word2vec, Pattern-idf

1 Introduction

We participated in the NTCIR-13 Short Text Conversation 2 (STC-2) Chinese subtask. Given a new post, this task aims to retrieve an appropriate comment from a large post-comment repository (Retrieval-based method) or generate a new appropriate comment (Generation-based method). Our system chooses the retrieval-based method.

The retrieved or generated comment for the new post is judged from four criteria: Coherent, Topically relevant, Non-repetitive and Context independent[]. The primary criterion for a suitable comment we consider is topically relevant. In other words, this comment should be talking about the same topic with the given post. In a previous work of retrieval-based STC, Ji et al.[] directly focus on the similarity between this new post and each comments. That is, when the new post and a candidate comment share same words or phrases, we can strongly infer that they may be on the same topic.

However, this is not a prerequisite of being a proper response of the post. For example, **(to do)** . In this case, post and comment is different in words but coherent. Such kind of comment is also non-repetitive.

To address this problem, we train Word2Vec[] model to obtain the similarity between new post and retrieved comment, LDA[?]) model and LSA[?]) model to obtain the degree of relateness. By combining them together, we proposed a simialrity score to search comment candidates, and achieves a good performance.

Based on a hypothesis, similar posts has similar corresponding comments, we try to find the similar posts to the new post and get their corresponding comments as a supplement for the candidates. In addition to Word2Vec model and LSA model, we also introduce langugae model trained by LSTM[**?**]) to compute post-post similarity.

In the last step, we rank the cadidates by Random Walk and pattern-idf respectively. Result shows that pattern-idf improve the performance while Random Walk deteriorate it instead.

The remainder of this paper is organized as follows: Section 2 describes our systems in detail. Our experimental results are presented in Section 3. We make conclusions in Section 4 .

2 System Architecture

The architecture of our system is described as Figure 1. It includes the following components.

2.1 Preprocessing

There are some traditional Chinese, specific symbols in raw text. We convert traditional Chinese to simplified Chinese and remove the specific symbols in order to clean the text.

Unlike English words in a sentence are separated by spaces, Chinese short texts are written without any symbol between characters. So the word segmentation becomes necessary. We choose nlpir[4] to segment the chinese text.

The following example in Table 3 shows the origin text, clean text and segentation result.

2.2 Similarity Features

In order to compute the degree of similarity or relateness between two sentences, we convert text sentence into continuous vector representations with some techniques including LSA, LDA, Word2Vec, LSTM.

LSA Latent semantic analysis (LSA) is a technique of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are close in meaning will occur in similar pieces of text (the distributional hypothesis). A matrix containing word counts per paragraph (rows represent unique words and columns represent each paragraph) is constructed from a large piece of text and a mathematical technique called singular value decomposition (SVD)

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

is used to reduce the number of rows while preserving the similarity structure among columns. We combine each post with its corresponding comments to be a document, then we train LSA model on these documents with gensim[5]. From trained model, we can get vector for each chinese word. Then, we can get vector representation of a sentence by Eq.1:

$$V = \frac{1}{n} * \sum_{1}^{n} v_i \tag{1}$$

Here, capital V refers to vector of a sentence, v refers to vector of each word in the sentence, and n is the length of the sentence.

LDA Latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. Like training LSA model, we combine each post with its corresponding comments to be a document, then we train LDA model on these documents with gensim. With trained LDA model, we can transform new, unseen documents into LDA topic distributions. We regard a sentence(a post, or a comment) as a document, convert it into plain bag-of-words count vector, and then index LDA model to obtain a vector representation of the sentence.

Word2Vec Word2Vec is an efficient tool for computing continuous distributed representations of words. We train our Word2Vec model on provided training text corpus with skip-gram architecture where window size is 10, vector length is 300 and min count is 20 to remove infrequent words. Vector representation for each chinese word is directly obtained from trained model. Then, we can get vector representation of a sentence by Eq.1.

LSTM ...

Cosine Similarity After convert sentence into vectors, we compute similarity of two sentences by cosine similarity:

$$Sim(s_1, s_2) = \frac{V_1 * V_2}{\|V_1\| \, \|V_2\|} \tag{2}$$

Here, V_1 refers to vector representation of sentence s_1, V_2 refers to vector representation of sentence s_2.

With sentence vectors from different models, we get corresponding sentence similarity. We use Sim_{LSA} to denote sentence similarity based on LSA model, analogously, Sim_{LDA} to denote sentence similarity based on LDA model and Sim_{W2V} to denote sentence similarity based on Word2Vec model.

[5] A python library for topic modelling, https://radimrehurek.com/gensim/index.html

2.3 Candidates Generation

Similar Posts Based on a hypothesis that similar posts has similar corresponding comments, we firstly find top-10 similar posts with a ranking score combining LSA, Word2Vec, and LSTM model:

$$Score_{q,p}(q,p) = Sim_{LSA}(q,p) * Sim_{W2V}(q,p) * Sim_{LSTM}(q,p) \tag{3}$$

Here, q denotes the query(the new post) p denote the post from repository.

Then, we get corresponding comments to the top-10 similar posts as first comment candidates, denoted as C_1.

Comment Candidats Since Word2Vec model captures semantic similarity, LSA or LDA reflects topic relateness, we combine LSA or LDA with Word2Vec respectively to directly retrieve top-N appropriate comments to the new post from all comments in the repository. N is equal to the number of comment candidates C_1.

$$Score^1_{q,c}(q,c) = Sim_{LSA}(q,c) * Sim_{W2V}(q,c) \tag{4}$$

$$Score^2_{q,c}(q,c) = Sim_{LDA}(q,c) * Sim_{W2V}(q,c) \tag{5}$$

Here, q denotes the query(the new post) c denote the comment from repository.

We combine the retrived top-N comments by Word2Vec model and LSA model with C_1 as final comment candidats for further reranking.

2.4 Reranking

Random Walk

Pattern-idf

3 Experiments

We submitted five runs for comparison and analysis:

1. Nders-C-R5: Use $Sim_{w2v} * Sim_{LDA}$ as a ranking score to directly retrieve and get top-10 comments from all comments.
2. Nders-C-R4: Use $Sim_{w2v} * Sim_{LSI}$ as a ranking score to directly retrieve and get top-10 comments from all comments.
3. Nders-C-R3:
4. Nders-C-R2:
5. Nders-C-R1:

4 Conclusions

In this paper, we

References