SRCB at the NTCIR-13 STC-2 Task

Li Yihan, Jiang Shanshan, Ding Lei,Tong Yixuan, Dong Bin
Ricoh Software Research Center Beijing Co., Ltd.
{Yihan.Li, Shanshan.Jiang, Lei.Ding, Yixuan.Tong, Bin.Dong}@srcb.ricoh.com

ABSTRACT
This is the first time SRCB participates in the Short Text Conversation (STC) task of Chinese. We developed conversation systems for both retrieval-based method and generation-based method. For retrieval based method, we proposed two models to retrieve post results from the repository based on the following three steps: preprocessing, candidate comment matching by indexing posts and comments in repository, and candidate comment ranking. For generation based method, we employed the state-of-the-art architecture Seq2Seq model to generate comments for posts. The evaluation results for both methods show that our proposed approaches achieve competitive results.

CCS Concepts
• Information systems→Information retrieval • Computing methodologies→Artificial intelligence.

Keywords
Retrieval; Generation; Comments; Neural Network

Team Name
srcb

Subtask
Short Text Conversation Task (Chinese)

1. INTRODUCTION
The task of Short Text Conversation (STC) is as follows: given a new post, the system is supposed to generate fluent, coherent and useful comments. The task provides participants a large amount of pairs of posts and comments from Weibo for Chinese task as training corpora.

At NTCIR-12, STC is taken as an IR problem [1]. The methods to solve IR problem try to find appropriate comments from corpora for posts in different ways. This year, at NTCIR-13, besides the retrieval-based method, generation-based method is also considered [2]. The generator can be modelled by using statistical machine translation model or the RNN-based neural model.

In this paper, we proposed approaches for both retrieval based method and generation based method. For the retrieval based method, our approach consists of the following steps: preprocessing, candidate comment matching by indexing posts and comments in repository, and candidate comment ranking. For generation based method, Sequence-to-sequence (Seq2Seq) with attention mechanism represents the state-of-the-art neural network model for comment generation. Thus, our generation based method is built on Seq2Seq model and consists of five parts: 1) Embedding; 2) Encoder; 3) Attention; 4) Decoder; 5) Beam Search. The two methods have been evaluated on STC-2 tests. The results indicate that both methods show competitive results.

2. Retrieval-based method
For retrieval-based method, we have set up two models for comments retrieval: S1 and S2 model.

2.1 S1: System Architecture
Given a new post, to retrieve top 10 comments from repository, preprocessing, matching and ranking are described as Figure 1.

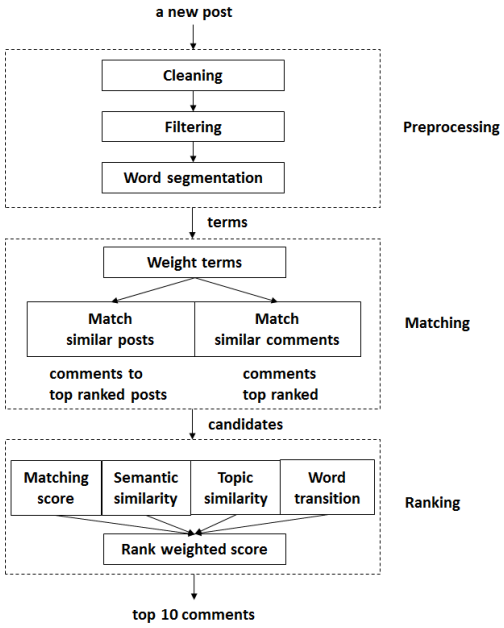

Figure 1. System overview.

2.1.1 Preprocessing
Since the post-comment pairs are collected from Weibo, although most noises are already removed, such as hashtags, external links, and forwards, there is still cleaning and filtering to do before retrieval. Firstly, punctuations and emoji are removed, then only Chinese characters, digits and letters are kept. Secondly, to normalize, upper cases are converted to lower cases, full-width characters are converted into half-width ones, traditional Chinese characters are converted into simplified ones. Finally, Jieba[1] is employed to segment a post/comment into words, with a 170k-word user dictionary.

2.1.2 Matching
Retrieval is a ranking problem. Obviously, to rank the whole repository for a new post requires too much computation which is unnecessary. Therefore, matching step is imported to narrow down the candidates to rank. During matching, both post-post similarity and post-comment similarity are considered. It is natural to assume that, two semantically similar posts may have the semantically similar comments, which means a comment from a similar post can be a good candidate to a new post. On the other hand, in the case of a post and a comment have more meaningful common words, they are very likely talking about the same topic which makes the comment a good candidate to the post.

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

According to above assumptions, given a new post denoted as q, matching includes the following steps:

1) Find top p similar posts to the new post from repository, and keep corresponding comments $\{c_p\}$ as candidates.
2) Find top r similar comments to the new post $\{c_r\}$ from repository, as candidates.
3) Candidate comment set is $\{c_p\} \cup \{c_r\}$.

To efficiently search a new post in repository, Apache Solr[2] is employed for index and query.

Furthermore, the posts and comments are segmented into word list in preprocessing. Simply, all the words of the new post can be considered as keywords. But, besides stop words which are always redundant semantically, words should have different weights in different context. More important words should be emphasized more in matching. We use TF-IDF based rules to weight terms, and take top weighted ones as keywords during search.

2.1.3 Ranking

The candidate comments are ranked by a weighted score. Firstly, a matching score $S_{match}(q,c)$ is defined as following:

1) If a candidate comment c comes from a similar post p, then $S_{match}(q,c) = relevance(q,p)$, where $relevance(q,p)$ is the relevance score of p.
2) If a candidate comment c is a similar to q, then $S_{match}(q,c) = relevance(q,c)$, where $relevance(q,p)$ is the relevance score of c.
3) If a candidate comment c meets the above two, then $S_{match}(q,c) = relevance(q,p) + relevance(q,c)$.

Secondly, a cosine similarity score $S_{cosine}(q,c)$ is calculated between the new post and a candidate comment. Suppose $\vec{q}$ and $\vec{c}$ are corresponding word vectors to q and c.

$$S_{cosine}(q,c) = \frac{\vec{q} \cdot \vec{c}}{\| \vec{q} \| \| \vec{c} \|}$$

Thirdly, a topic similarity score $S_{topic}(q,c)$ is calculated between the new post and a candidate comment. Suppose $\overrightarrow{q_t}$ and $\overrightarrow{c_t}$ are corresponding topic vectors to q and c. Topic vectors (5~30 dimension) are trained by topic modeling.

$$S_{topic}(q,c) = \frac{\overrightarrow{q_t} \cdot \overrightarrow{c_t}}{\| \overrightarrow{q_t} \| \| \overrightarrow{c_t} \|}$$

Fourthly, anther semantic similarity score $S_{word2vec}(q,c)$ is calculated between the new post and a candidate comment. Suppose $\overrightarrow{w_v}$ is corresponding distributional representations to a word w.

$$S_{word2vec}(q,c) = \frac{1}{2}\left(\sum_{w_1 \in q} \max_{w_2 \in c} distance(w_1,w_2) + \sum_{w_1 \in c} \max_{w_2 \in q} distance(w_1,w_2)\right)$$

$$distance(w_1,w_2) = \frac{\overrightarrow{w_1} \cdot \overrightarrow{w_2}}{\| \overrightarrow{w_1} \| \| \overrightarrow{w_2} \|}$$

Fifthly, a transition score $S_{transition}(q,c)$ from the new post to a candidate comment is calculated. Suppose $tp[w_1][w_2]$ is the transition probability from a post word w_1 to a comment word w_2 which is statistics from repository.

$$S_{transition}(q,c) = \sum_{w_1 \in q} \sum_{w_2 \in c} (tfidf(w_1) * tfidf(w_2) * tp[w_1][w_2])$$

Finally, a weighted sum of above five scores are used to rank out top 10 comments.

2.2 S2: System Architecture

The architecture of our S2 system includes the following 3 components: Preprocessing, Candidate Generation and Candidate Ranking.

Instead of search engine, we user Apache Spark[3] to calculate a similarity degree of each post-post pair and post-comment pair. The similarity degree is used in candidate generation and ranking.

2.2.1 Preprocessing

The preprocessing of our method contains two parts: Chinese conversion and word segment.

For the Chinese conversion part, traditional Chinese and converted into simplified ones. The common symbols are all converted into English form and excess punctuations are removed to obtain clean text.

For the word segment part, Jieba was used to split the Chinese text into a sequence of words. Besides, extract dictionary was loaded to ensure a higher accuracy. The dictionary mainly contains names extracted from Wikipedia[4].

2.2.2 Similarity Calculation

Benefit from the arithmetic capability of Apache Spark, we calculated a similarity degree of each post-post pair and post-comment pair instead of indexing the text and retrieving with search engine.

The similarity degree was calculated as the cosine similarity of the vectors of the two texts. The text was represented with so-called Vector Space Model [3].

The weight of each word in the VSM representation contained three parts:

1) TF-IDF value of the word.
2) Language model [4] importance of the word: the difference of the probabilities for the text to be a nature language with and without the word.
3) POS weight: nominal words with higher weight.

2.2.3 Candidate Generation

Given a post, the candidate comments contained two parts:

1) The top-10 comments that have higher similarity with the given post.
2) The corresponding comments of the top-10 posts that have higher similarity with the given post

2.2.4 Candidate Ranking

We trained a decision tree regression model to predict the score of the comment with the given post. The regression model only contains three features:

1) The similarity between the comment and given post.

[2] http://lucene.apache.org/solr/

[3] http://spark.apache.org/

[4] https://zh.wikipedia.org/

2) The similarity between the given post and the corresponding post of the comment that has highest similarity with the given post.

3) The predict score of a sequence to sequence model [5] we trained.

The regression model is trained on the labeled training dataset.

2.3 Experiments

For S1 model, we submitted 4 runs for comparison and analysis. For S2 model, we submitted 1 run.

1) srcb-C-R1: Only use $S_{match}(q,c)$ and $S_{cosine}(q,c)$ as a naive baseline system.

2) srcb-C-R2: Besides $S_{match}(q,c)$ and $S_{cosine}(q,c)$, $S_{topic}(q,c)$ is imported.

3) srcb-C-R3: Besides $S_{match}(q,c)$ and $S_{cosine}(q,c)$, $S_{transition}(q,c)$ is imported.

4) srcb-C-R4: Besides $S_{match}(q,c)$ and $S_{cosine}(q,c)$, $S_{word2vec}(q,c)$ is imported.

5) srcb-C-R5: Import language model importance and generation-based model score in retrieve.

The reason why we didn't give all five scores a chance to work together is because we found them performing poorer than the baseline (srcb-C-R1). Besides, we find that language model importance and generation-based model score do bring a little promotion in ranking (Mean nG@1).This is also proved in official STC results as shown in Table 1.

Table 1. Official STC results for team srcb

Run	Mean nG@1	Mean P+	Mean nEER@10
srcb-C-R1	0.4343	**0.5395**	**0.5736**
srcb-C-R2	0.3972	0.5030	0.5368
srcb-C-R3	0.3852	0.4964	0.5272
srcb-C-R4	0.2983	0.4306	0.4688
srcb-C-R5	**0.45**	0.5367	0.5644

3. Generation-based method

3.1 System Architecture

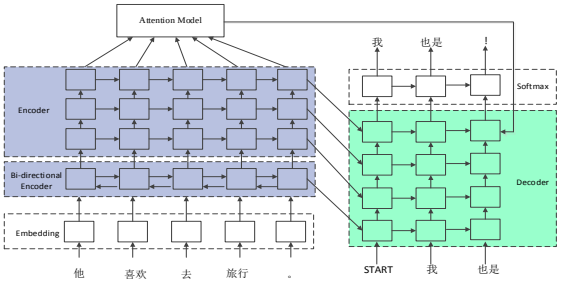

Figure 2. System Architecture of our generation-based method.

Recent research work has shown that neural network based models have yielded impressively flexible results in conversation contexts. Sequence-to-sequence (Seq2Seq) [6] with attention mechanism [7] represents the state-of-the-art neural network model for comment generation. As with traditional Seq2Seq model, our system consists of five components. Figure 2 shows the system architecture of our generation-based method.

1) Embedding: Word in sentences are mapped to vectors of real numbers as inputs to neural layers.

2) Encoder: The encoder transforms source sentence into a list of vectors, which represents the meaning of all word read so far.

3) Attention model: The attention model allows decoder to focus on different regions of source sentence.

4) Decoder: Calculates the probability of the next symbol given the source sentence encoding and the decoded target sequence so far.

5) Beam search: Search comments and rank the comments.

3.2 Preprocessing

The original data set for model training were extracted from the Internet, thus there could be plenty of arbitrarily expressions. We took the following steps to clean up these data:

1) Convert the traditional Chinese words to Simplified Chinese words.

2) Convert the DBCS (double byte character set) cases to the SBCS (single byte character set) cased using ZHConverter[5]. The SBC cases include punctuations and numbers.

3) Delete the Unicode Emoji marks.

4) Replace English words with its lower case.

5) Delete short sentences consisting less than 3 words.

6) Delete sentences with no Chinese words.

7) Normalize recurring punctuations with unified marks.

8) Replace numbers with a mark to represent a number.

These steps could eliminate noisy characters and merge characters with similar meanings together, so that sentences could be expressed with less information loss using less vocabulary. For the training data set, commonly comments were also removed. These preprocessing stages above were aimed for model developing data sets and word embedding training corpuses. Jieba was hired for Chinese sentence segmentation.

3.3 Model Configuration

Our conversation model support several configurations of parameters for neural network. Table 2 list the detailed configurations.

To train model with high performance, we have tries different configurations of parameters. In our final submitted runs, we used some fixed values for some parameters, as these values show better performance for the neural model. The initial learning rate is set to 1 as training data are large and this would accelerated training speed. The decay of learning rate is set to 0.99, and once the current loss is the maximum in the last five updates, the decay is invoked. The encoder consists of one bi-directional LSTM and three single directional LSTM, while the decoder consists of four single directional LSTM. Regularization factor is set to 5. SGD is used to optimize all the parameters. We used the attention function proposed by Bahdanau et al. [5]. Other parameters are variant for our submitted runs.

[5] https://github.com/program-in-chinese/zhconverter

Table 2. Supported configuration of parameters

Parameters	Desc	Values
Learning Rate	The extent of gradient descent	Real number
Decay of LR	Update LR	Real number
Strategy of LR	Strategy for updating LR	Algorithm design
Gradient Descent	Algorithm of gradient descent	SGD,Adagrad , …
Layer	number of layers for network	[1-10]
Number of node	Number of node	128,256,…
Batch_Size	Batch size used for gradient update	32，64，128,…
LSTM/GRU	Type of RNN	LSTM,GRU
Regularization Factor	Prevent overfitting	Real number
Dropout Rate	Dropout Rate	Real number
Attention Function	Correlation between encoder and decoder	Algorithm design

3.4 Refining Comments

Using Beam search in the output stage could generate multiple comments for every post sentence. However, these comment suffered greatly from two issues: First, it is very likely to result in similar replies with only one or two words different. Second, the model could repeat words or phrases in results. Rules were designed to beat these problems:

1) To start with, beam search should be modified to produce more candidates. In our system, 100 raw comments was listed for each post.
2) Use edit distance to measure the similarity between two sentences, and divide the similarity score by the length of the shorter one for normalization. Remove similar candidates according to the normalized similarly, so that the remaining comments were different from each other. Two sentences were regarded similar if their similarity score was below 0.5.
3) A repeat score was introduced to measure the extent of repeat. It was obtained by computing the percentage of words appear more than once among all words within a sentence. Candidates were removed if their repeat score exceed 0.4.

Other comments consisting meaningless or dirty words were also removed.

3.5 Experiments

We have submitted 5 runs for generation-based model. The setting of each run is described as followings:

1) SRCB-C-G1 used word as basic splitting unit for sentences, and the basic setting is describes in subsection 3.3.
2) SRCB-C-G2 is almost the same as G1, except that it sets dropout rate to 0.2 for both encoder and decoder.
3) SRCB-C-G3 is almost the same as G2, except that it prunes some post-comment pairs whose comments occur frequently in the training sets and uses the remaining pairs as training data.
4) For the run SRCB-C-G4. A 3-layered Bi-LSTM model was chosen. The dimension for word2vec was 512. The batch size was 128. The size of vocabulary was 80,000. Dropout was skipped in this run. Other parameters were the same with SRCB-C-G1.
5) For the run SRCB-C-G5. An 8-layered Bi-LSTM character-based model was chosen. The dimension for pertained char vector was 512. The batch size was 128. The size of vocabulary was 10,000. Dropout was skipped in this run. Other parameters were the same with SRCB-C-G1.

The results of our submitted five runs is shown is Table 3. The metrics used is same to retrieval-based methods.

As we can see from the table, when nG@1 and P+ metrics are considered, SRCB-C-G2, which incorporate dropout rate, can improve the evaluation measures significantly. Compared with SRCB-C-G3 shows competitive results with SRCB-C-G2. When nERR@10 is considered, SRCB-C-G3 shows best results in all runs.

Table 3: Official STC results for our submitted runs

Run	nG@1	nERR@10	P+
SRCB-C-G1	0.3160	0.4582	0.4997
SRCB-C-G2	**0.4138**	0.5188	**0.5782**
SRCB-C-G3	0.4103	**0.5269**	0.5737
SRCB-C-G4	0.3657	0.4838	0.5241
SRCB-C-G5	0.3052	0.4376	0.4735

4. Conclusions

In this paper, we propose two approaches which rely on retrieval-based method and generation-based method separately for STC Chinese task of NTCIR-13. For retrieval based method, we propose two models to retrieve post results from the repository based on three steps. For generation based method, we employ the state-of-the-art architecture Seq2Seq model to generate comments for posts. The evaluation results for both method show that our proposed approaches achieve competitive results.

5. REFERENCES

[1] Lifeng Shang, Testsuya Sakai, Zheng Dong Lu, Hang Li, Ryuichiro Higashinaka, Yusuke Miyao. Overview of the NTCIR-12 Short Text Conversation Task, Proceedings of NTCIR-12, 2016.

[2] Lifeng Shang, Testsuya Sakai, Hang Li, Ryuichiro Higashinaka, Yusuke Miyao, Yuki Arase, Masako Nomoto. Overview of the NTCIR-13 Short Text Conversation Task, Proceedings of NTCIR-13, 2017.

[3] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." Communications of the ACM 18.11 (1975): 613-620.

[4] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[5] Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[6] Sutskever, I., Vinyals, O., and Le, Q.V. 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, 3104-3112.

[7] Luong, M. T., Pham, H., and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.