USTC at NTCIR-12 STC Task

Junbei Zhang
University of Science and Technology of China, China
zjunbei@mail.ustc.edu.cn

Junfeng Hou
University of Science and Technology of China, China
hjf176@mail.ustc.edu.cn

Shiliang Zhang
University of Science and Technology of China, China
zsl2008@mail.ustc.edu.cn

Lirong Dai
University of Science and Technology of China, China
lrdai@ustc.edu.cn

ABSTRACT

In this paper, we describe the system submitted by USTC team for the Short Text Conversation (STC) task of the NTCIR-12. We proposed transition-p2c, encoder-decoder-Reverse and joint-Train models for the STC task and submitted 5 official runs. The transition-p2c model provides transition probability between post and comment in word's level which complements the TF-IDF feature. The encoder-decoder-Reverse and joint-Train model provide semantic similarity between post and comment. With the help of these models, we achieved 0.2867 on Mean nDCG@1, 0.4509 on Mean P+ and 0.4181 on Mean nERR@10.

Team Name

USTC

Subtasks

STC (Chinese)

Keywords

end-to-end, encoder-decoder, learning to rank

1. INTRODUCTION

Dialogue is one of the most challenging NLP tasks. To build a traditional dialogue system which contains several components[1], a lot of related technologies have been developed such as dialogue state tracking[2], natural language generation[3] and so on. Meanwhile a very popular approach recently developed is to train end-to-end models with recurrent neural network on a large amount of real dialog transcripts[4, 5]. However, the end-to-end models lack goal-oriented frameworks and are difficult to evaluate.

Although large amount of work have been done, the progress of conversation between human and computer is still quite limited. To improve this situation, the Short Text Conversation (STC) task[6], which is one of the NTCIR-12 pilot tasks, is proposed. In the STC task, participants are given a dataset of the post-comment pairs crawled from Weibo, and requested to find the most suitable comment from this dataset when a new post comes in. Some labeled post-comment pairs are also provided.

Some work related to STC task have been done before the task becomes an NTCIR-12 pilot task. Wang[7] and Ji[8] collected a large amount of conversation data and defined the task as an information retrieval (IR) problem. Shang[4] used end-to-end model to generate responses.

As an NTCIR-12 pilot task, the STC task is taken as an IR approach of the one round short Text Conversation. And the evaluation measures are Mean nDCG@1, Mean P+ and Mean nERR@10, which are popular in IR field. We model the task as a learning-to-rank problem as Wang [7] and Ji[8] do. All the works we have done here are to find efficient and accurate matching features that can help the ranking model to distinguish proper responses from improper ones.

Our contributions can be listed as follows:

1) Consider the transition probability of post to comment in word's level.

2) Use end-to-end encode-decode model not only for post to comment (EncDec-Forward, for short), but also for comment to post (EncDec-Reverse, for short).

3) Combine EncDec-Forward and EncDec-Reverse model together.

2. SYSTEM ARCHITECTURE

We model the task as a learning-to-rank problem as Shang [4] do. In order to get a good ranking list, we must employ representative features that can be used to train the ranking model. To deal with this, we classify the features into two categories : lexical features and semantic features. The lexical features help to select the responses similar to the posts in word's level while the semantic features help to select the most semantic relevant responses.

2.1 Lexical feature

2.1.1 Query-Response and Query-Post Similarity

We use vector space model to get the similarity between a query and a post as well as a query and a response. The vector of the query or post is their own TF-IDF score vector. All the formulas are the same with [8]. The word segmentation tool is *Jieba*[1] Chinese text segmentation and the TF-IDF is calculated by *gensim*[2]. All the words are reserved and unigram is used.

This work was supported in part by the Science and Technology Development of Anhui Province, China (Grants No. 2014z02006) and the Fundamental Research Funds for the Central Universities (Grant No. WK2350000001).

[1]https://github.com/fxsjy/jieba
[2]https://radimrehurek.com/gensim/index.html

Algorithm 1 Transition-p2c Train

```
Input: repos-post, repos-comment
Output: transition matrix $\mathbf{T}$
1: Word segmentation, and get words' vector of post and comment
2: Initialize: $\mathbf{T}$ = zeros(m, n), m = length of post vocabulary set, n = length of comment vocabulary set
3: IDF score of post set and comment set
4: for (p, c) in (post-word vector, comment-word vector) do
5:     $Get\ tf-idf\ score\ vector$ : p-tf-idf, c-tf-idf
6:     $\mathbf{T} = \mathbf{T} + \textbf{p-tf-idf} \cdot \textbf{c-tf-idf}^T$
7: end for
8: Normalization: for i in [0, m], normalize $\mathbf{T}$[i]
```

Algorithm 2 Transition-p2c Test

```
Input: test-query, repos-comment
Output: transition score
1: Initialize: score = 0, $\mathbf{K}$ = zeros(m, n)
2: Get tf-idf score vector of test-query and comment
3: $\mathbf{K} = \textbf{query-tf-idf} \cdot \textbf{c-tf-idf}^T$
4: for (m, n) in $\mathbf{K}$.shapes do
5:     $score = score + \mathbf{K}[m][n] * \mathbf{T}[m][n]$
6: end for
```

2.1.2 *Transition-p2c*

Besides the Query-Response and Query-Post similarity, there is still the transition probability between post words' vector and comment words' vector. For example, when the post contains 'New Year', then the comment mostly contains 'Happy'. So the transition probability from 'New Year' to 'Happy' should be high.

The modeling details of the transition probability between post words' vector and comment words's vector (transition-p2c, for short) can be found from Algorithm 1 and 2.

2.2 Semantic feature

2.2.1 *EncDec-Forward and EncDec-Reverse model*

Seq2seq model is a well-known end-to-end neural network model[9, 10]. The model encodes an input sentence with recurrent neural network and decodes an output sentence. The model can learn reasonablely semantic mapping and relations in MT task[9] and in response generation task[4]. Motivated by the work in [9] and [4], we use the seq2seq model to estimate the likelihood of a response given a post. And the likelihood (normalized with sentence length) is used as query-response similarity measure.

The input sentences in this model are usually posts, and the output sentences are usually comments similar to [4]. We call it encDec-forward model for short. However, unlike machine translation, one post in STC may have several comments to fit with, and one comment can also fit with more than one post. They are ambiguous and multi-modal. By maximizing P(comment|post), the model may only learn to predict the most frequent utterances as shown in [11]. So we treat comment as input and post as output and maximize P(post|comment), which requires the comment to contain more information and to decide which post is the most likely one. And it can be expected that the ***frequent utterance*** phenomenon may appear less common. We call it

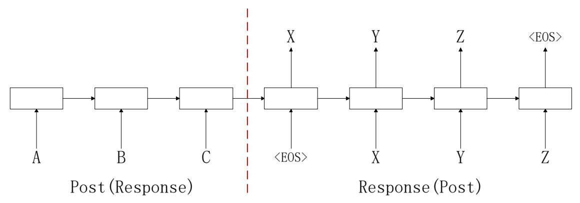

Figure 1: Encoder-Decoder Model for STC

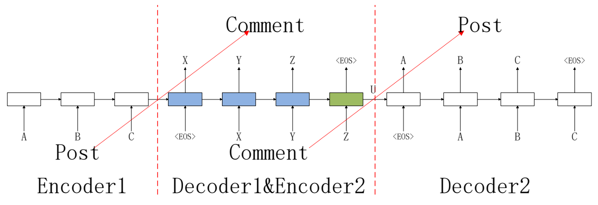

Figure 2: Joint-Train Model

encDec-reverse model for short.

In STC task, we can get the post and comment, and caculate the likelihood of one given another. Obviously, both of the P(post|comment) and P(comment|post) should be considered. We also tried attention model but no gain in training set. So we just use the encoder-decoder model without attention, as shown in figure 1.

2.2.2 *Joint-Train model*

We also tried to combine the modeling of P(post|comment) and P(comment|post) in one model (we call it joint-Train model). The architecture of joint-Train model is as figure 2 shows. Decoder1 is regarded as both a decoder and an encoder(encoder2). Different from the normal encoder-decoder model, another decoder(decoder2) is added.

Firstly, we use encoder1 to encode the post, with comment as decoder1's target, which means the neural network is required to use post to predict comment. The objective function of this part is denoted as O_1. By multiplying the last layer output of decoder1(the green block in figure2) and the transition matrix U, a joint representation of post and comment is formed. Then we use this representation as the input of decoder2, with post as decoder2's target, which means the neural network is asked to use post and comment to reconstruct post. The objective function of this part is denoted as O_2. Finally, the objective function of the joint-Train model is the weighted average of O_1 and O_2.

As we can see, the encoder1 and the decoder1 models the transition of post to comment, while the encoder2(decoder1) and the decoder2 models the transition of comment to post. Here decoder1 is equivalent to be employed as encoder2, which means the parameters of them are shared.

2.3 Ranking

After all the features are obtained, we need a ranking model to merge all the scores and output a final score for each query and response pair. The ranking list is generated by the final ranking score. In STC task, we use linear RankingSVM [12]. The model is trained by the given labeled post-comment pairs and is the same as [8].

3. EXPERIMENTS AND ANALYSIS

3.1 Implementation Details

First of all, we use Query-Response Similarity, Query-Post Similarity, Transition-p2c and EncDec-Reverse Model to get 10000 pairs each from the whole repository, and remove the same pairs. After that, we can get a candidate repository around 37000 pairs. The hyperparameters of EncDec-Forward and EncDec-Reverse model are: batch_size = 1024, hidenLayer_dim = 1024, word_dim = 150, word_vocab = 4000(covers 99.7% of total words). And the hyperparameters of JointTrain model are: batch_size = 512, hidenLayer_dim = 1024, word_dim = 100, word_vocab = 4000, loss function = 0.2 * post-comment loss(encoder1&decoder1, O_1) + 0.8 * comment-post loss(encoder2&decoder2, O_2).

3.2 Results

The evaluation measures in STC task are mean nDCG@1, mean P+ and mean nERR@10. We submit 5 runs, and the file names are:

- USTC-C-R1: Query-Response Similarity + Query-Post Similarity + EncDec-Forward + EncDec-Reverse + Transition-p2c
- USTC-C-R2: Query-Response Similarity + Query-Post Similarity + EncDec-Forward + EncDec-Reverse + JointTrain
- USTC-C-R3: Query-Response Similarity + Query-Post Similarity + EncDec-Forward + Transition-p2c
- USTC-C-R4: Query-Response Similarity + Query-Post Similarity + EncDec-Forward + EncDec-Reverse
- USTC-C-R5: Query-Response Similarity + Query-Post Similarity + EncDec-Forward

The results in the task are shown in table 1. These systems are selected offline based on their performance on training set. The training set performance is shown in table 2. As we can see, the training set performance is very different from test set. Unfortunately, the baseline system R5 yields best result on test set.

The main diffierence between online and offline evaluation is the subset selection. When submit a top10 comment list for a query in online test set, we should search the whole repository. However, a very small subset has already been given when we evaluate our system offline with labeled data. And we only need to rank the subset for each query rather than search the whole repository.

In order to analyze the effectiveness of the features we proposed, we will give some cases and see whether they can improve the ranking order.

Table 1: Official STC(Chinese) results

Run	nDCG@1	P+	nERR@10
R5	**0.2867**	**0.4509**	0.4160
R4	0.2767	0.4479	**0.4181**
R1	0.2733	0.4499	0.4169
R2	0.2567	0.4310	0.4001
R3	0.2267	0.4094	0.3848

Table 2: STC(Chinese) training set results

Run	nDCG@1	P+	nERR@10
R5	0.4741	0.6529	0.6327
R4	0.4785	0.6582	0.6395
R3	0.4726	0.6570	0.6347
R2	**0.4889**	**0.6625**	0.6446
R1	0.4859	0.6618	**0.6449**

3.3 Case Study

To get a better analysis about the models employed in our system and to complement the experimental results above, we show some cases about the models. It should be pointed out that the cases are good ones chosen by human.

3.3.1 Transition-p2c feature

Transition-p2c models the transition probability of the words from post to comment. We rank post-comment word pairs by transition score from high to low, and remove the pairs whose post word and comment word are the same. Table 3 shows the top10 pairs. As we can see, transition probability can model lexical relations between different words which TF-IDF can not.

Table 3: Transition score top10 of different word pairs

post words	comment words	transition score
运费 (freight)	代购 (purchasing agents)	0.3207
中型(medium)	谢谢(thanks)	0.1302
警报(alarm)	口水(saliva)	0.1273
元宵节 (Lantern Festival)	快乐 (happy)	0.1260
萌到(sprout)	可爱(lovely)	0.1180
拜年 (pay a New Year call)	新年快乐 (happy new year)	0.1177
王老吉 (Wong Lo Kat)	加多宝 (JDB Beverage)	0.1077
本地(native)	流量(traffic)	0.1066
小家伙(kiddy)	可爱(lovely)	0.1042
张国荣 (Leslie Cheung)	哥哥 (brother)	0.1007

3.3.2 EncDec-Reverse feature

As explained before, encDec-Reverse feature can help the system to select more suitable comments instead of frequent utterances. An example is given in table 4. Combining with the encDec-Reverse feature (R4), the model lifts the good comment from order 7(R5) to 1(R4) and lowers the bad comment from order 8(R5) to 12(R4).

3.3.3 Joint-Train feature

Although the encDec reverse feature can suppress the frequent comment, sometimes it may introduce some noise or even totally wrong comments. So it is necessary to train the encDec forward and reverse jointly. An example is given in table 5.

For the query in table 5, the first comment can be lowered by the encDec reverse, with ranking from 3 to 10. But the

Table 4: Case for encDec-Reverse feature

query	
晚上加餐了，在宾馆旁边喝羊汤 Having soup near the hotel this evening as an extra meal	
(label)comment	rank change
(L1)给你加餐 嘎嘎哟 ~ Offer an extra meal for you, haha~	8 -> 12
(L2)比我这个呆学校无法加餐的娃幸福多了 I have to stay at school without extra meal which makes you much happier than me	7 -> 1

Table 5: Case for joint-Train feature

query			
开始下雪了，长达半年的漫长冬天又开始了 Snowing now. The long winter lasting for half year begins			
(label)comment	R5	R4	R2
(L1)好漫长漫长漫长漫长。。 So long, so long...	3	10	5
(L0)以前是只喜欢夏天,但最近开始想念冬天了 I like summer only before, but I begin to miss winter recently	56	7	14

The R5 system is TF-IDF+encDec-Forward. The R4 system is TF-IDF+encDec-Forward+encDec-Reverse. The R2 system is TF-IDF+encDec-Forward+encDec-Reverse+joint-Train.

second comment is lifted from 56 to 7 which is harmful. By introducing the joint-Train, we can lower the first comment from 3 to 5 while not lifting the second one too much from 56 to 14. This example shows the effectiveness of the joint-Train. From the example we can also see that the reverse part affect the ranking order a lot. This may result from that the reverse part has a bigger weight when we train the model jointly. How to set a proper weight is an issue, and we will explore it in future work.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed transition-p2c, encDec-Reverse and jointTrain model for the STC task. The results in training set and cases shown the efficiency of the models, although the online evaluation is inconsistent with the offline evaluation because of the subset selection problem.

To imporve the performance of our system, several future works are planned as follows:

- Deep encoder-decoder model. Single layer of one input vector may not be enough for catching the high level features. So to try deep encoder-decoder modeling may be helpful.
- Find a better way to joint EncDec-Forward and EncDec-Reverse model together.
- Study the consistency of the online and offline evaluation for STC task.

5. REFERENCES

[1] James H Martin and Daniel Jurafsky. Speech and language processing. *International Edition*, 2000.

[2] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, 2013.

[3] Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*, 2015.

[4] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015.

[5] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[6] Lifeng Shang, Tetsuya Sakai, Zhengdong Lu, Hang Li, Ryuichiro Higashinaka, and Yusuke Miyao. Overview of the NTCIR-12 short text conversation task. In NTCIR, 2016.

[7] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. A dataset for research on short-text conversations. In *EMNLP*, pages 935–945, 2013.

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

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[10] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[11] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*, 2015.

[12] T Joachims. Training linear SVMs in linear time proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD). 2006.