CIAL at the NTCIR-13 STC-2 Task

Yung-Chun Chang
Graduate Institute of Data Science, Taipei Medical University, Taiwan
changyc@tmu.edu.tw

Yu-Lun Hsieh
Institute of Information Science, Academia Sinica, Taiwan
morphe@iis.sinica.edu.tw

Wen-Lian Hsu
Institute of Information Science, Academia Sinica, Taiwan
hsu@iis.sinica.edu.tw

ABSTRACT

Short text conversation (STC) has emerged as a prominent research topic and gained considerable attention in recent years. While it is still an open problem whether the retrieval-based method should be replaced by or combined with generative models for STC task, the NTCIR-13 STC-2 Task provides a transparent platform to compare the two aforementioned methods via doing comprehensive evaluations. In this task, we proposed a retrieval-based method with distributed vector representation, and a generation-based method with recurrent neural networks. Overall, we submitted 4 and 1 official runs for retrieval and generation settings, respectively. We also proposed a data augmentation method for extending the amount of labeled data that is more sufficient for training a generative model.

Keywords

embeddings, SVM, RNN

Team Name

CIAL

Subtasks

Chinese subtask with two different settings: retrieval-based method and generation-based method

1. INTRODUCTION

Natural language conversation is one of the most challenging artificial intelligence problems, which involves language understanding, reasoning, and the utilization of common sense knowledge. Recently, due to the explosive growth of microblogging services such as Twitter and Weibo, the amount of conversation data available on the web has tremendously increased. Moreover, with the emergence of social media and the wide spread of mobile devices, conversation via short texts has become an important way of communication (called short text conversation, STC). Many real-life applications can benefit from the research on STC, for instance, automatic message reply on mobile phone, voice assistants like Siri, and various chatbots for use with smart home devices.

Instead of multiple rounds of conversation, STC only considers one round of conversation, in which each round is formed by two short texts, with the former being an input (referred to as post) from a user and the latter a output given by the computer (referred to as response).

There are two categories in NTCIR-13 STC-2 [6], 1) the retrieval-based method (RBM), and 2) the generation-based method (GBM). RBM is taken as an information retrieval problem by maintaining a large repository of post-comment pairs from Weibo, and then finding a clever way to reuse these existing comments to respond to new posts. On the other hand, the purpose of GBM is to generate new comments. In essence, it treats the response generation as a translation problem, in which the model is trained on a parallel corpus of post-response pairs. The difference in these two modes are illustrated in Figure 1.

In NTCIR-13 STC-2, we participated Chinese short text conversation subtask in both task settings (i.e., RBM and GBM). Finally, We submitted 4 and 1 official runs for RBM and GBM setting, respectively. In this paper, we described the algorithms, tools and resources used in CIAL shot text conversation system.

2. SYSTEM ARCHITECTURE

The system architecture of our proposed method is comprised of two subsystems, namely, a distributed representation-based response retrieval model and a generation model based on augmented training set, as shown in Figure 2. The retrieval-based approach is rooted from calculating similarity between input text and text of labeled post, and retrieve the most appropriate comment as response. On the other hand, we based on recurrent neural network to develop a sequence to sequence model for generating responses. The following subsections describe both methods in detail.

2.1 Retrieval Model

One simple approach to STC is to take it as an information retrieval (IR) problem, maintain a large repository of short text conversation data, and develop a conversation system mainly based on IR technologies. As shown in the Figure 2, our retrieval-based method is comprised of three key components, namely, distributed representation for short text, post linking, and response ranking. Given a message, the system retrieves related responses from the repository and returns the most reasonable response. More specific, we select the most suitable response as reply to the current message without generating a new response. Formally, for a given message M, we select from the repository of post-comment pairs (P_r, C_r) the response C^* with the highest ranking score.

$$\begin{aligned} C^* &= \arg\max sim(M, P_i), \text{where} \\ i &= (P_r, \arg\max C_r) \end{aligned} \quad (1)$$

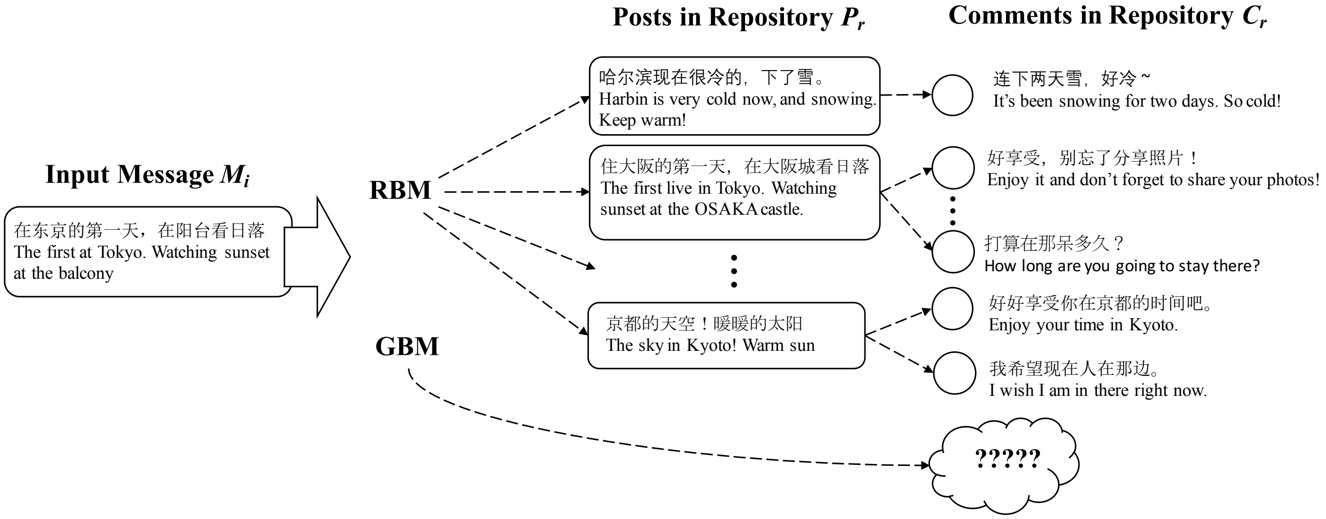
Posts in Repository P_r
Comments in Repository C_r
Input Message M_i
在东京的第一天，在阳台看日落
The first at Tokyo. Watching sunset at the balcony
RBM
GBM
哈尔滨现在很冷的，下了雪。
Harbin is very cold now, and snowing. Keep warm!
住大阪的第一天，在大阪城看日落
The first live in Tokyo. Watching sunset at the OSAKA castle.
京都的天空！暖暖的太阳
The sky in Kyoto! Warm sun
连下两天雪，好冷~
It's been snowing for two days. So cold!
好享受，别忘了分享照片！
Enjoy it and don't forget to share your photos!
打算在那呆多久？
How long are you going to stay there?
好好享受你在京都的时间吧。
Enjoy your time in Kyoto.
我希望现在人在那边。
I wish I am in there right now.
?????

Figure 1: Retrieval-based and generation-based methods in producing appropriate response to an input message.

At the outset, we learned distributed representation (embeddings) of every character in the corpus by merging all posts and comments and utilizing the program `word2vec`[1] [5]. The embeddings are used to transform words into real-valued vectors for use in subsequent components in the model. The dimension of embeddings is 300.

For the training process, each post P_i in the repository is transformed into a distributed vector representation using pre-trained character embeddings. P_i is represented as an average of character sequence to train an SVM using `libsvm` [1]. To retrieve a comment as response for the M_i, it is first transformed into a distributed vector representation as well. The post linking component then adopt the trained classifier to find a target post P^* that is the most similar to M_i. However, there are several comments for P^* in the training data, and each comment has three annotators to score $(+1, 0, -1)$ the degree of suitable for P^*. We thus further rank comments through summation of annotation scores. Finally, a comment (i.e., C^*) with highest annotation score is chosen as the response for M_i.

In addition, we employ the retrieval model to extend the training corpus for use in the generation model. This process begins at finding similar comments in addition to the provided ones. Recall that the official labeled corpus contains post-comment pairs, where the same post can be linked to multiple comments with different scores. We refer to a group of post-comment pairs, in which the posts are identical, as a post set $\mathbb{P}$. We then utilize the post linking method in the proposed retrieval model to find additional comments that are similar to the ones in a $\mathbb{P}$. Identically, we can find posts that closely resembles the one in a $\mathbb{P}$. These newly-obtained posts and comments are randomly paired to augment the original labeled corpus.

2.2 Generation Model

The corpus for training the generation model is an augmented dataset described in the previous section. We adopt the sequence-to-sequence model [8] with attention mechanism [4] for generating comments. The input is a sequence of characters that constitutes the post content, and the output is also a sequence of characters that is a coherent response to the post. Our model mainly consists of: embedding layer, recurrent encoder layers, attention layer, and recurrent decoder layers. The embedding layer is initialized with the same embeddings used in the retrieval model described above. The recurrent encoder layer is composed of a bidirectional LSTM [3, 2], with recurrent states of 300 dimensions as well. The attention layer employs the Global Attention mechanism [4] to calculate weights over the entire output sequence of the encoder depending on decoder output, and produces another sequence of vectors for the decoder. Finally, the decoder layer is also comprised of (uni-directional) LSTMs with 300-dimension recurrent states. They receives the output from attention layer and computes a sequence of V-dimensional vector where V equals to the number of possible characters, and we can infer the corresponding character from these vectors so as to from a readable string of characters. We implemented the sequence-to-sequence generation model using `pytorch`[2].

3. RESULTS AND DISCUSSION

Performances are evaluated on three metrics, i.e., nG@1 (normalised gain at cut-off 1), P+, and nERR@10 (normalized expected reciprocal rank at cutoff 10) [7].

Table 1 lists the official results of the proposed methods. Our submission, prefixed by 'CIAL', contains four runs from the retrieval model (denoted as R1–R4) and one from the generation model (denoted as G1). R1 to R4 represents different settings of the threshold in the retrieval engine. All of them obtained unsatisfactory scores. However, the generative model performs better than some settings of the retrieval one. We postulate that, although the generative model can be successful in other domains, it may require much more data or better pre-training strategy in order to improve the outcome.

4. CONCLUSIONS

[1] https://code.google.com/archive/p/word2vec/

[2] https://github.com/pytorch/pytorch

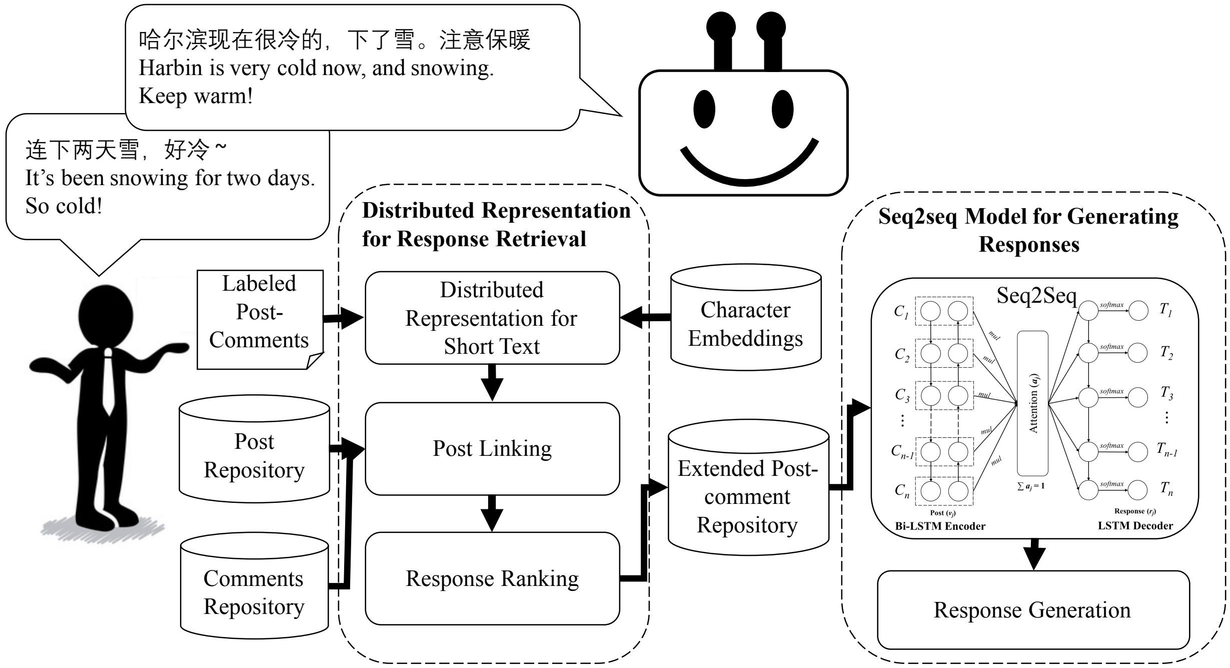

Figure 2: System architecture of the proposed method.

Table 1: Results for CIAL systems at STC-2 Task.

System	Mean		
	nG@1	P-plus	nERR@10
CIAL-C-R2	0.0769	0.1386	0.1354
CIAL-C-R1	0.0769	0.1244	0.1129
CIAL-C-R4	0.0022	0.0265	0.0238
CIAL-C-R3	0.0022	0.0152	0.0108
CIAL-C-G1	0.0797	0.0981	0.0748
Mean	0.2736	0.3590	0.3916

In this paper, we proposed a short text conversation system with two different models at NTCIR-13 STC-2 Task, Overall, we submitted 4 run results for retrieval-based method, and 1 run result for generation-based method. We also devise a novel data augmentation method in order to create a larger corpus for training of the generation model. In the future, we will explore different ways to integrate deeper semantics and syntactic information into short text. Moreover, we will also utilize keyword extraction approach to filter out redundant text for improvement of our distributed representation.

Acknowledgments

This work was supported by grant MOST106-3114-E-001-002 and TMU106-AE1-B16 from the Ministry of Science and Technology in Taiwan and Taipei Medical University, respectively.

5. REFERENCES

[1] C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

[2] A. Graves, S. Fernández, and J. Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*, pages 753–753, 2005.

[3] S. Hochreiter and J. Schmidhuber. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479, 1997.

[4] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[6] L. Shang, T. Sakai, H. Li, R. Higashinaka, Y. Miyao, Y. Arase, and M. Nomoto. Overview of the NTCIR-13 short text conversation task. In *NTCIR*, 2017.

[7] L. Shang, T. Sakai, Z. Lu, H. Li, R. Higashinaka, and Y. Miyao. Overview of the NTCIR-12 short text conversation task. In *NTCIR*, 2016.

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