iNLP at the NTCIR-13 STC-2 Task

Long Qiu *
Onehome (Beijing) Network Technology Co. Ltd.
qiulong@onehome.me

Jingang Wang
Alibaba Group
jingang.wjg@alibaba-inc.com

Sheng Li
Alibaba Group
lisheng.ls@alibaba-inc.com

Junfeng Tian
East China Normal University
jftian@stu.ecnu.edu.cn

Jun Lang
Alibaba Group
langjun.lj@alibaba-inc.com

ABSTRACT

The iNLP team participated in the Short Text Conversation (STC) task of NTCIR-13. This report describes our attempts to solve the STC problem and discusses the official results.

Team Name

iNLP

Subtasks

Short Text Conversation Subtask (Chinese)

Keywords

Short Text Conversation, Information Retrieval, Chatbot, Variational Autoencoders

1. INTRODUCTION

The iNLP team participated in the Short Text Conversation (STC) task of NTCIR-13. This report describes our approaches to the STC problem and discusses the official results.

To have a machine capable of carrying out meaningful conversations with humans is one of the most challenging problems of NLP, yet it attracts increasing attention from both industry and academia. We share the view that, for their potential to efficiently and effectively interact, engage, and therefore serve people, chatbots will play a increasingly important role in our daily life.

We focus on the Chinese subtask of NTCIR-13 STC-2. Formally, STC is framed as a single round, therefore simplified natural language conversation task: the goal is for the machine to response to a human post by a fluent comment with *relevant*, *coherent*, *self-sustained* and *substantial* information. The official evaluation measures include nG@1 (normalized Gain at cutoff 1), P+, and nERR@10 (normalized Expected Reciprocal Rank at cutoff 10). We investigate both methods suggested by the organizer: Retrieval-based and Generation-based.

For the retrieval-based method, our strategy is to maintain a large repository of short text conversation data, and retrieve candidate comments given posts based on the latest IR technologies. Since STC-1 of NTCIR-12 only consider STC as an IR problem, most of the participating systems adopt the classical architecture consisting of retrieval, matching and ranking components [6]. We also borrow this architecture for our retrieval-based methods. More concretely, we attempt different sets of matching features after retrieving candidate comments for target posts and perform supervised ranking strategies to obtain the final ranking list. The results illustrate the potential of our retrieval-based method.

For the generation-based method, we adopt the RNN-based sequence to sequence model. It encodes the input post into a vector, a representation, and outputs the desired comments based on it [12]. When the idea is applied to STC, it is known as a Neural Responding Machine (NRM) [10]. While the information in the final hidden state of the encoder seems sufficient, the preceding hidden states contain additional, complementary information that could be harnessed by the attention mechanism [1] to further enhance the overall performance. Compared with SMT, NRM is shown to generate more fluent and semantically relevant comments. For STC-2, one of our main interests is how NRM could produce multiple comments for a given post. Variational autoencoders (VAEs) [8] are selected for this end. A VAE encodes input into a latent variable, which allows us to inject information to achieve diversified comments. Although diversity is not yet the focus of STC (at least not for the evaluation), we believe it is crucial for a chatbot to have the ability to be creative, to come up with different valid comments when presented with even repeated posts.

2. RETRIEVAL-BASED APPROACH

This section presents our retrieval-based methods for Chinese STC, including unsupervised and supervised ones. Figure 1 presents the framework of our system, and the unsupervised component is enclosed in the box.

Given a post, the system retrieves related comments from a repository of post-comment data and returns the most suitable ones. We mainly rely on the Weibo [1] conversation corpus provided by the organizer (STC-2 corpus) as our post-comment repository. Formally, for a given post q, we rank the candidate comments in the repository according to the relevance score involving q and a post-comment pair

*Corresponding author

[1] http://weibo.com/

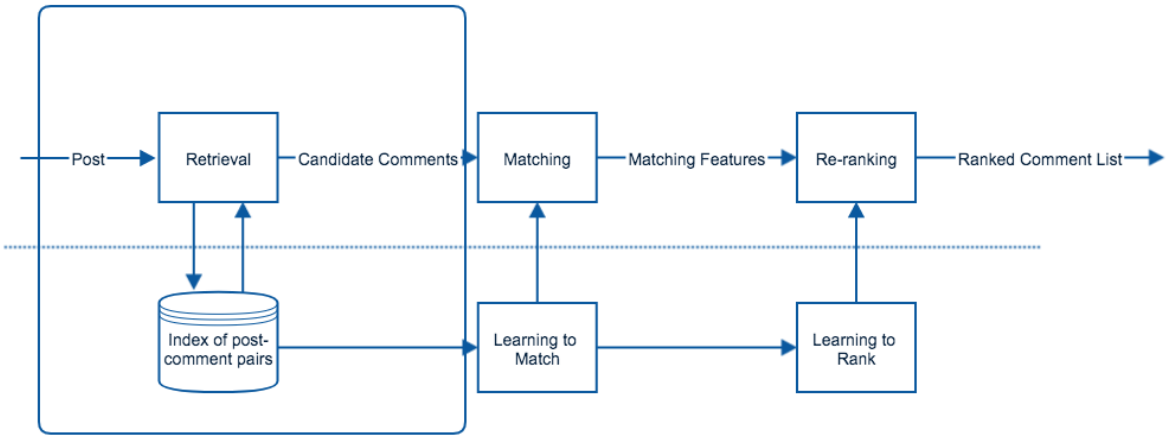

Figure 1: Retrieval-based Framework

(p, c):

$$score(q, (p, c)) = \sum_{i=1}^{N} \omega_i \phi_i(q, (p, c)), \quad (1)$$

where ω_i is the weight of the i-th matching feature $\phi_i(\cdot)$, and N is the total number of matching features.

2.1 Preprocessing

Since the Weibo posts and comments are crawled from social websites, the texts are not all well-formed. Preprocessing is prerequisite for further analysis. We perform Chinese word segmentation to split the posts and comments into individual words, using the toolkit jieba[2], a popular open source Chinese word segmentation tool. In addition, we remove stopwords and emoticons before indexing.

2.2 Index

In order to process the voluminous post-comment pairs efficiently, we resort to ElasticSearch[3], an open-source Lucene-based text search engine, to index the whole corpus.

In addition to STC-2 corpus, we also acquire STC-1 corpus and another public Weibo corpus [10] (denoted as ACL' 15) to further expand the repository. The three corpora are partially overlapped. Table 1 demonstrates the statistics of them in detail. Please note that the post-comment pairs are many-to-many mapping in the corpora.

We build the indices with two types (i.e., post or comment) for STC-1 and STC-2 respectively. With this setting, given a post (or a comment), we can retrieve relevant post or comment flexibly.

2.3 Unsupervised Ranking

Given a post, one can issue a search against the index, and ElasticSearch would return a candidate comment list ranked by relevance score, which is adopted by BUPT team [13] in STC-1. We take this approach as our baseline unsupervised approach (i.e., iNLP-C-R2 in Table 2).

The relevance score of ElasticSearch is a variant of BM25 [9] based on TF-IDF-like calculation. Besides, cosine similarity based on word embeddings between the target post and candidate comments can be used as the ranking metric. We employ gensim word2vec[4] to learn word embeddings with the whole repository. The approach of iNLP-C-R3 is an unsupervised baseline, which ranks the candidate comments according to their cosine similarity with the given post.

We also enhance the word2vec-based approach with query expansion, a technique widely used in Information Retrieval, which is useful in improving system recall performance. Given a post, we retrieve its relevant posts from the post repository and combine them as a new query to search candidate comments from the comment repository. This query expansion method is noted as iNLP-C-R1 in Table 2.

2.4 Supervised Ranking

For our supervised ranking model, linear RankSVM [5] is used to learn the weights in Equation 1. iNLP-C-R5 is based on such a model with all the matching features proposed in [6] except the translation-based features. We also submit a result where relevance score is also included in the feature set (iNLP-C-R4).

One difficulty of supervised methods is the lack of training data. As demonstrated in Table 1, the labeled pairs are not enough to train robust rankers, considering that the ranking label is a three-point-scale metric (i.e., L2, L1 and L0 in STC-2). This may be the reason that our supervised methods cannot beat the unsupervised ones.

Thus we submit 5 retrieval-based runs in total, and their details are shown in Table 2. Note that all the comments in our retrieval-based submissions are retrieved from STC-2 comment index solely, although we include STC-1 data to improve the ranking performance.

3. GENERATION-BASED APPROACH

In a typical neural encoder-decoder framework, the input $\mathbf{x_i}$ is transformed into a hidden representation $\mathbf{h_i}$, upon which the decoder produces the desired output $\mathbf{y_i}$:

$$\mathbf{h_i} = f(\mathbf{x_i}, \boldsymbol{\phi}) \quad (2)$$

$$\mathbf{y_i} = g(\mathbf{h_i}, \boldsymbol{\theta}). \quad (3)$$

[2] https://github.com/fxsjy/jieba
[3] https://www.elastic.co/
[4] https://radimrehurek.com/gensim/models/word2vec.html

Table 1: Statistics of STC-1, STC-2 and ACL' 15 corpora.

	STC-1	STC-2	ACL' 15	Total
post #	196, 395	219, 174	219, 276	368, 027
cmnt #	4, 637, 926	4, 305, 706	4, 307, 678	7, 984, 674
post-cmnt pairs #	5, 648, 128	4, 345, 193	4, 347, 176	9, 675, 324
labeled pairs #	6, 016	11, 535	N.A.	N.A.

Table 2: Retrieval-based Submissions of iNLP team.

Runs	Description
iNLP-C-R1	Rank the comments via word2vec-based similarity with query (post) expansion
iNLP-C-R2	Rank the comments with ElasticSearch relevance score
iNLP-C-R3	Rank the comments via word2vec-based similarity without query (post) expansion
iNLP-C-R4	RankSVM-based reranking with STC-1 and STC-2 training data (whole feature set)
iNLP-C-R5	RankSVM-based reranking with STC-1 and STC-2 training data (without ElasticSearch match feature)

Table 3: Generation-based Submissions of iNLP team.

Methods	Description
iNLP-C-G1	**LDA_post**: RNN Model with Attention and VAE Decoder + 1st post topic, $z_r \sim Norm(0.0, 0.9)$
iNLP-C-G2	**LDA_comment**: RNN Model with Attention and VAE Decoder + 1st comment topic, $z_r \sim Norm(0.0, 0.9)$
iNLP-C-G3	**Random**: RNN Model with Attention and VAE Decoder, $z_r \sim Norm(0.1, 0.8)$
iNLP-C-G4	**Random**: RNN Model with Attention and VAE Decoder, $z_r \sim Norm(0.0, 0.8)$
iNLP-C-G5	**LDA_comment**: RNN Model with Attention and VAE Decoder + 2nd comment topic, $z_r \sim Norm(0.0, 0.9)$

[12] is such a framework successfully applied to Machine Translation (MT). MT is naturally a one-to-one mapping problem: for a given sentence in the source language, we normally expect a single most suitable counterpart in the target language as the translation. However, this does not always hold in STC. In many cases, an online post is responded with many different, yet equivalently relevant comments. As it is unclear what a valid hidden representation should be, except for the fact that it encodes pertinent information required by a good translation, it is not clear how to achieve diversity by manipulating $\mathbf{h_i}$. We can sample from the distribution of $\mathbf{y_i}$, but the quality of the resultant texts is out of control. Fortunately, there exists more attractive alternatives, some permit more control over the input to the decoder.

As introduced by [8], VAEs encode the input $\mathbf{x_i}$ into a hidden representation, too. However, this hidden representation, known as latent variable $\mathbf{z_i}$, can be expected to follow a (multivariate) Gaussian distribution $p(\mathbf{z}) \sim Norm(\boldsymbol{\mu}, \boldsymbol{I})$. So while we have similarly

$$\mathbf{z_i} = f(\mathbf{x_i}, \boldsymbol{\phi}), \tag{4}$$

and

$$\mathbf{y_i} = g(\mathbf{z_i}, \boldsymbol{\theta}), \tag{5}$$

the loss function for each training instance is now:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{x_i}, \mathbf{y_i}) = &- E_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x_i})}[\log p_{\boldsymbol{\theta}}(\mathbf{y_i}|\mathbf{z})] \\ &+ KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x_i})||p(\mathbf{z})),\end{aligned} \tag{6}$$

where the first term on the RHS is the reconstruction loss, and the second term KL divergence is a regularizer. Once we applied the reparameterization trick, $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ can be learnt via standard optimization algorithms such as SGD.

Theoretically, after the model is learnt, one could sample directly from the variational distribution $p(\mathbf{z}) \sim Norm(\boldsymbol{\mu}, \boldsymbol{I})$ and evoke only the decoder to generate an output, in the STC case, a comment. But in the STC scenario, multiple comments is expected to be the responses of a given post. Therefore, we include the input post in the picture. Employing both the encoder and the decoder of the VAE model, we experiment with three different settings toward comment diversity for a post $\mathbf{x_i}$:

- **Random**: We first sample value $\mathbf{z_i} \sim p_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x_i})$, then add some random disturbance $\mathbf{z_r} \sim Norm(\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r)$. In stead of $\mathbf{z_i}$, the decoder has $\mathbf{z_i'}$ as input:

$$\begin{aligned}\mathbf{y_i} &= g(\mathbf{z_i'}, \boldsymbol{\theta}) \\ &= g(\mathbf{z_i} + \mathbf{z_r}, \boldsymbol{\theta}).\end{aligned} \tag{7}$$

- **LDA_post**: Similar to Zero-shot translation [7], we combine the input post $\mathbf{x_i}$ with its most probable topic predicted by LDA [2]. We treat this topic information as an extra token like a regular word token but from a vocabulary for topics. After doing this, we sample $\mathbf{z_i}$ as in **Random** setting:

$$\begin{aligned}\mathbf{x_i^{topic}} &= LDA(\mathbf{x_i}), \\ \mathbf{z_i} &= f([\mathbf{x_i^{topic}}, \mathbf{x_i}], \boldsymbol{\phi}), \\ \mathbf{y_i} &= g(\mathbf{z_i'}, \boldsymbol{\theta}) \\ &= g(\mathbf{z_i} + \mathbf{z_r}, \boldsymbol{\theta}).\end{aligned} \tag{8}$$

- **LDA_comment**: Alternatively, we inject topic information based on the most probable LDA topic of the comments suggested by our retrieval-based approach.

We employ 3-layer RNN with LSTM cells for both the encoder and decoder, with their individual parameters. The size of each layer is 256. The meta-parameter k (number of topics) for LDA is 300. The input tokens are individual Chinese characters or sequences of digits and letters. After normalizing numbers, we keep the most frequent 7000 such tokens in the vocabulary. For inference, the decoder generates outputs in a greedy manner. The resultant comment is never longer than the post. In most cases, it ends when

Table 4: Official STC-2 Chinese Results of iNLP Runs (out of 120 runs)

Run	Rank	Mean nG@1	Run	Rank	Mean P+	Run	Rank	Mean nERR@10
SG01-C-G1	1	0.5867	SG01-C-G1	1	0.6670	SG01-C-G1	1	0.7095
SG01-C-R1	4	0.5355	SG01-C-R3	4	0.6200	SG01-C-R3	4	0.6663
iNLP-C-R1	26	0.4132	iNLP-C-R1	20	0.5375	iNLP-C-R1	25	0.5674
iNLP-C-R2	33	0.4055	iNLP-C-R2	24	0.5324	iNLP-C-R2	27	0.5667
iNLP-C-R4	43	0.3790	iNLP-C-R4	38	0.5025	iNLP-C-R4	38	0.5408
iNLP-C-R3	46	0.3695	iNLP-C-R3	47	0.4899	iNLP-C-R3	45	0.5264
iNLP-C-G4	68	0.2477	iNLP-C-G2	67	0.3579	iNLP-C-G2	67	0.3911
iNLP-C-G2	70	0.2323	iNLP-C-G4	69	0.3490	iNLP-C-G4	68	0.3839
iNLP-C-G1	71	0.2320	iNLP-C-G5	71	0.3414	iNLP-C-G1	70	0.3732
iNLP-C-G5	72	0.2257	iNLP-C-G1	72	0.3411	iNLP-C-G3	72	0.3672
iNLP-C-G3	73	0.2227	iNLP-C-G3	73	0.3344	iNLP-C-G5	73	0.3654
iNLP-C-R5	74	0.2187	iNLP-C-R5	74	0.3142	iNLP-C-R5	74	0.3291

Table 5: Translated Sample iNLP Outputs (for test-post-10850)

Test Post	The Italian pianist played so beautifully on the Norwegian Arctic glacier. The world stopped to listen!
iNLP-C-R1	In fact, Norway is the best country!
iNLP-C-R1	This is Norway [whining]. Super beautiful.
iNLP-C-R1	This is Norway! The workmanship of nature!
iNLP-C-G2	I want to go to Beijing. But not so much.
iNLP-C-G2	If only I could actually go there.
iNLP-C-G2	I want to go. But I don't like it.

the decoder outputs the first <EOS>. The randomness we introduced by adjusting the latent variable $\mathbf{z}$ renders comments of multiple versions. To certain extent they are iid, therefore it is unnecessary to further rank on them.

We submit 5 generation-based runs, and the details are shown in Table 3. Our generation-based model is trained only on the STC-2 corpus. There are 219,174 posts and 4,305,706 corresponding comments. After removing comments with less than 5 tokens, we end up with 4,100,960 post-comment pairs. Among them, 10,000 randomly sampled pairs are allocated as the development set and the rest form our training set. The 11,535 comments with manually labels are not used here.

4. EVALUATION

STC evaluation assesses the appropriateness of the comments. Human assessors assign a score to each comment based on four criteria: 1) Fluent, 2) Coherent, 3) Self-sufficient, and 4) Substantial. Both 3) and 4) are required for a score L2, and both 1) and 2) are required for a score of at least L1. Three evaluation measures are considered while comparing the performance of different participants: nG@1, $P+$, and $nERR$@10.

Table 4 shows the scores of our submissions. We also include the best performing retrieval-based run (SG01-C-G1) and generation-based run (SG01-C-R1) for comparison. For a full list, we refer the reader to the official STC-2 report [11].

Our best performing run is iNLP-C-R1, the retrieval-based approach utilizing word2vec-based similarity with query expansion. Among 120 submitted runs, it ranks top 20%. The effectiveness of query expansion is demonstrated by the extra gain it makes (compared to iNLP-C-R3). Meanwhile, the ElasticSearch baseline, iNLP-C-R2, is surprisingly rather strong. In comparison, the RankSVM-based runs (iNLP-C-R4/5) are not performing as well as one would expect, which we suspect is due to the limited training data. It is worth noticing that ElasticSearch match feature does seem to help here.

Our generation-based runs generally are no match to their retrieval-based peers, a trend manifests at STC-2. Also, their scores are rather close, suggesting that the injected topics are overlooked by the decoder. Nonetheless, those scores hint too much digression is not a good idea (iNLP-C-G3 vs. iNLP-C-G4), and the most relevant topics are the most probable topic of the comment (iNLP-C-G2), followed by the most probable topic of the post itself (iNLP-C-G1).

We show some sample outputs of our retrieval-based and generation-based runs (iNLP-C-R1 and iNLP-C-G2) in Table 5. When the retrieval algorithm finds comments highly relevant to the post, they are fluent, coherent and normally self-sufficient, as they are comments composed by actual online-users. However, substantiality is something we can only hope for, since our retrieval or ranking algorithms does not particularly emphasize it. In contrast, even the comments given by the generation-based approach can sometimes be arguably substantial (*If only I could actually go there*), they are not as fluent, coherent or self-sufficient as those of the retrieval-based approach.

There are in total 15577 tokens, 1425 types in Run iNLP-C-R1, and 11361 tokens of 652 types in Run iNLP-C-G1. Similar gap exists between other retrieval-based runs and generation-based runs. The retrieval-based approach prefers longer comments with a larger vocabulary. In contrast, the generation-based approach restricts itself with a set of short comments that are apparently safe to apply everywhere (high likelihood). To address this problem, other loss functions should be considered.

5. CONCLUSION

We attempt both retrieval-based and generation-based approaches to the Short Text Conversation (STC) task of NTCIR-13. The retrieval-based approach features a carefully designed pipeline: a search engine (ElasticSearch) first retrieves a set of candidate comments, then word2vec-based textual similarity measure ranks these comments by their similarity with the input post. In addition, this pipeline is shown to benefit from an optional query expansion component.

We also believe supervised approaches such as Learning to Match and Learning to Rank would be suitable for STC. However, with limited training data, our implementation does not rank the candidate comments as favourably as the unsupervised approach does.

Our generation-based approach is essentially a VAE, where the introduction of a hidden variable gives us an opportunity to generate comments with certain diversity, yet at the same time still highly related to the input post. We also propose to integrate semantic topics to further control the desired diversity, and we show one way to integrate LDA topics by treating them as special input tokens. We are interested in VAEs because it is natural to inject controlled semantic diversity with them. Therefore, they are highly suitable for the scenario of STC where a variety of responses could be equally plausible, addressing different aspects of the original post, or even contributing new content to the conversation. Unfortunately, the resultant comments are not quite satisfactory. We hope a much richer corpus and different loss functions for the decoder could improve some of the desired qualities.

VAEs are meant for robust encoding, so noise is intentionally added to the intermediate hidden variable. When applied to images, they are therefore known to generate "blurred" output. Generative Adversarial Nets [4], on the contrary, generates images with resolution high enough to confuse a trained discriminator. But it does not allow generation from a particular post, let along a response with certain degree of coverage of one. It thus would be exciting to see how a combination of them [3] can perform at STC.

Other possible directions for future work are: to learn explicitly how human conversations switch from one semantic topic to another, and to learn how to predict suitable follow-up topics with that knowledge. Perhaps a more intuitive way to inject new topics into the conversation is to train a bunch of different encoders, each with its own field of interests.

6. REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representationsm (ICLR)*, pages 1–15, 2015.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[3] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially Learned Inference. pages 1–18, 2016.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. pages 1–9, 2014.

[5] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. 2000.

[6] Z. Ji, Z. Lu, and H. Li. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*, 2014.

[7] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation.

[8] D. P. Kingma and M. Welling. Stochastic Gradient VB and the Variational Auto-Encoder. In *Proceedings of the 2nd International Conference on Learning Representationsm (ICLR)*, pages 1–14, 2014.

[9] S. Robertson, H. Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[10] L. Shang, Z. Lu, and H. Li. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1577–1586, 2015.

[11] 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 *Proceedings of NTCIR-13*, 2017.

[12] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014.

[13] Y. Tan, M. Wang, and S. Han. Buptteam participation in NTCIR-12 short text conversation task. In *Proceedings of NTCIR-12*, 2016.