# Gbot at the NTCIR-13 STC-2 Task

Hainan Zhang[†,‡], Tonglei Guo[†,‡], Yanyan Lan[‡], Jiafeng Guo[‡], Jianing Li[†,‡] and Xueqi Cheng[‡]
[†]University of Chinese Academy of Sciences
[‡]CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology
{zhanghainan, guotonglei, lijianing}@software.ict.ac.cn, {lanyanyan, guojiafeng, cxq}@ict.ac.cn

## ABSTRACT

This paper describes our participation in STC-2 Chinese subtask of NTCIR-13. All runs are submitted for both two tasks, namely generation-based task and retrieval-based task. Various methods based on Seq2Seq framework were used to generate the responses. Interaction-focused method based on deep learning models is used to deal with relevance between queries and comments. As for generation-based task, we propose to add a constraint term to the original objective function of Seq2Seq, which further incorporates the coherent between post and reply into consideration. Specifically, three different approaches have been adopted to measure the similarity between post and reply, i.e. unlearned constraint such as cosine function and pretrained matching function such as bilinear function and MatchPyramid. As for information retrieval task, we apply a method to obtain relevance that is crucial to retrieval. As called interaction-focused method, we apply method based on MatchPyramid, which first builds interactions between a query and a comment and then uses a deep model to obtain the representation for the interactions and the relevance score.

## Team Name

Gbot

## Subtasks

Short Text Conversation Task(Chinese)

## Keywords

STC, deep learning, seq2seq, generation

## 1. INTRODUCTION

Short text conversation task is a significant problem in natural language processing. The STC-2[10] has two new pilots tasks: generation-based task and retrieval-based task. Generation-based models are trained by learning to predict the next word given the input dialogue posts and its generated history. From human's perspective, there are two criteria about what is a good reply: fluency and coherency. On one hand, the reply must be a normal and fluency sentence that meets the standard of everyday language. On the other hand, the reply must be in correlation with the post, which makes it look like a human's real reply. Thus we conduct three kinds of constraints to guide the generation to be close to the post.

As for generation-based models, most existing works follow the Seq2Seq[13] framework. It uses an LSTM encoder to encode the input post into a vector representation, as an initial state of decoder, and then decode the ground-truth reply with LSTM decoder using the maximum likelihood estimation(MLE) objective function. However, the generated reply of Seq2Seq is fluent but not related enough. It is easy to generate safe and general reply sentences, such as '我们都是这样的(We are all like this)' and '这个是什么意思啊?(What is it?)'. Li et al.[4, 5] also found that Seq2Seq is easy to generate general replies like 'I do not know' and 'Yes, it is'.

The constraints of post and generation can characterize the degree of topic proximity. In dialogue generation process, we can evaluate the proximity degree of post and generation as a constraint score at the end of the generation, and back-propagate the constraint score to each cell of the LSTM decoder. The constraints can guide the model to generate the sentence, which is in accordance with the post.

In this model, we propose three kinds of constrained models for the post and generation:

- Intuitively, we use an unlearned constraint as cosine similarity function(SIM) for the reason that we think the post and generation may share a same topic.
- We use a pretrained matching function, MatchingPyramid function, as pretrained constraint. The post and reply may share some keywords in their reply. Thus we use a match pyramid model[8] as a constraint(MP) to model this situation.
- We use a pretrained matching function, BiLinear function, as pretrained constraint. We use a bi-linear model[12] as a constraint(BL) to model the transferred topic situation.

For retrieval-based task, the task can be seen as an Information Retrieval problem. We focused on the methods using deep models to find relevance between queries and comments. The problem can be formalized as a matching problem.

We apply MatchPyramid in interaction-focused models. Firstly convert the given query and the comment to evaluate into embedding vectors, then conduct an interaction matrix by calculating cosine similarity between each post word embedding vector and each comment word embedding vector. Then use the matrix as input for a CNN network with max-pooling to get representations vectors for the interaction, finally using a MLP to obtain the matching score for the given pair of the query and the comment.

# 2. GENERATION-BASED METHOD

## 2.1 Sequence to Sequence

Our first model is the typical LSTM-based Seq2Seq framework [1] used in dialogue generation. Given a post $X = \{x_1, \ldots, x_M\}$ as the input, a standard LSTM first maps the input sequence to a fixed-dimension vector $h_M$. Then another LSTM is used as the decoder to map the vector $h_M$ to the ground-truth response $Y = \{y_1, \cdots, y_N\}$. Typically, the decoder is trained to predict the next word $g_i$, given the context vector $h_M$ and the previous generated words $\{g_1, \ldots, g_{i-1}\}$. In other words, the decoder defines a probability over the output $Y$ by decomposing the joint probability into the ordered conditionals by chain rule in the probability theory:

$$\begin{aligned} P(Y|X) &= \prod_{i=1}^{N} p(y_i|h_M, y_1, \ldots, y_{i-1}) \\ &= \prod_{i=1}^{N} g(h_M, y_{i-1}, h'_i), \end{aligned} \tag{1}$$

where $g$ is a softmax function, $h'_i$ is the hidden state in the decoder LSTM.

Usually the attention mechanism is further introduced to the above Seq2Seq framework in real applications. Instead of using $h_M$ as the context vector in the decoder, we let the context vector, denoted as $s_i$, to be dependent on the sequence $(h_1, \cdots, h_M)$. Each $h_k$ contains information about the input sequence with a strong focus on the parts surrounding the $k$-th word of the input sentence. The context vector $s_i$ is then computed as a weighted sum of these $h_k$:

$$s_i = \sum_{k=1}^{M} \alpha_{ik} h_k. \tag{2}$$

The weight $\alpha_{ik}$ of each representation $h_k$ is computed by:

$$\begin{aligned} \alpha_{ik} &= \frac{\exp(e_{ik})}{\sum_{j=1}^{M} \exp(e_{ij})}, \\ e_{ik} &= v^T \tanh(W_1 h'_{i-1} + W_2 h_k), \end{aligned} \tag{3}$$

where $v^T$,$W_1$ and $W_2$ are learned parameters. $e_{ik}$ is an alignment model which scores how well the inputs around position $k$ and the output at position $i$ match. The score is based on the LSTM hidden state $h'_{i-1}$ (just before emitting $y_i$), and $h_k$ of the input sentence.

Given a set of training data $\mathcal{D}$, Seq2Seq assumes that data are i.i.d. sampled from a probability $P$, and use the following negative log likelihood as the objective for minimization.

$$\mathcal{L} = -\sum_{(X,Y)\in\mathcal{D}} \log P(Y|X). \tag{4}$$

## 2.2 Constraints

We think the constraints of post $X$ and generation $G$ should be used in the conversation generation model. For the reason that the constraints of post and generation can not take a derivation, we multiply the constraints as a weight to the loss function like Li et al.[5, 6, 14] did in their work:

$$\mathcal{L}_m = -\sum_{(X,Y)\in(D)} \mathrm{cons_m}(X,G) \times \log P(Y|X) \tag{5}$$

We propose three kinds of constraints and describe them in detail as following. The first constraint SIM is an unlearned method, a direct calculation method without any model parameters. Then the second constraint is pre-trained matching model. And the third constraint is an end-to-end training model.

### 2.2.1 Embedding Similarity

Our second model is based on the similarity of post and generation embedding as a simple constraint. We define the cosine similarity of $X$ and $G$ as a constraint function.

$$\mathrm{cons_{SIM}}(X,G) = 1 - cosine(Average(X), Average(G)) \tag{6}$$

where $Average(X)$ is an embedding which is the mean over the word embeddings in sentence $X$ and $Average(G)$ is an embedding in $G$. The $cosine(Average(X), Average(G))$ is a kind of method to evaluate the topic similarity of $X$ and $G$.[9, 11]

We use the $\mathrm{cons_{SIM}}(X,G)$ as a constraint weight constraint$(X,G)$ to guide the reply generation process.

### 2.2.2 Matching Pyramid

Our third model is based on the MatchingPyramid model[8]. We use a match pyramid model, a good matching ranking list model, as a constraint to model this situation. Given the post $X$ and the generation $G$, we use a match pyramid model to calculate the score:

$$s_{mp}(X,G) = \mathrm{Matching} - \mathrm{Pyramid}(X,G) \tag{7}$$

We pre-train this model using the pairwise loss function. In training step, we randomly select five negative replies as negative samples. Thus in order to keep the accordance of the training and testing, in testing step, we randomly select five negative generated sentences $\{GN_1^{mp}, \ldots, GN_5^{mp}\}$ and define the Matching Pyramid constraints:

$$\mathrm{cons_{MP}}(X,G) = 1 - \frac{s_{mp}(X,G) - mn}{mx - mn} \tag{8}$$

where $mx$ and $mn$ is the max and min score of the score set $[s_{mp}(X,G), s_{mp}(X,GN_1^{mp}), \ldots, s_{mp}(X,GN_5^{mp})]$.

### 2.2.3 BiLinear

Our fourth model is based on the BiLinear model[12]. Given the post $X$ and the generation $G$, we use a GRU[2] model to encode both as embedding $em(X)$ and $em(G)$. And we add a Bi-Linear transfer:

$$s_{bi}(X,G) = em(X) \times W \times em(G) \tag{9}$$

where $W$ is a matrix of the transformation.

We trained this model using the pairwise loss function[3]. In training step, we randomly select five negative replies as negative samples. So in order to keep the accordance of the training and testing, we randomly select five negative generated sentences $\{GN_1^{bi}, \ldots, GN_5^{bi}\}$ and define the BiLinear constraints as:

$$BL(X,G) = 1 - \frac{s_{bi}(X,G) - mn}{mx - mn} \tag{10}$$

where $mx$ and $mn$ is the max and min score of the score set $[s_{bi}(X,G), s_{bi}(X,GN_1^{bi}), \ldots, s_{bi}(X,GN_5^{bi})]$.

We use the $BL(X,G)$ as a constraint weight to guide the reply generation process.

## 2.3 Experiments

For the Seq2Seq-att[1], and our three constrained models, we set the number of RNN hidden nodes as 300, batch size as 200. All the models share the word embeddings trained on the STC2 training data for STC2 Dataset.[1] We adopt the gradient decent method with learning rate 0.5 rather than Adam, because it has better performance in our experiments. We decrease learning rate with the decay factor 0.99, when the training loss continuously increases through three iterations. Firstly, we directly use character level sentence as input rather than word level sentence, because word segmentation in Chinese is not perfect and words are much sparser than characters, which leads to poor performance in Chinese dialogue generation. Then, we set the vocabulary size as 5000 which have high frequency and set '<UNK>' for the unknown words. We also set '0' for all the number in STC2 data.

To show the constrained models can generate more diverse replies which are more like a human real reply, we use the degree of diversity [4, 5] by calculating the number of distinct unigrams and bigrams in generated responses. The distinct-unigram is defined as:

$$dist\_uni = \frac{len(U)}{\sum_{w \in U} size(w)}$$

where $U$ is the unigram set of generate words and $size(w)$ is the number of word $w$ generated by the model. And distinct-bigram is similar to the definition of distinct-unigram:

$$dist\_bi = \frac{len(B)}{\sum_{w \in B} size(w)}$$

where $B$ is the bigram set of generated words and $size(w)$ is the number of bigram words $w$ generated by the model.

To show the constrained models' responses share semantic similarity to the post and ground-truth response, we consider three similarity metrics based on word embeddings in the same way as Serban et.al[9]. The Embedding Average of post(Post-Average) metric is the same as Eq.(6). And the Embedding Average of reply(Reply-Average) metric is

$$cosine(Average(G), Average(Y))$$

This metric is widely used for measuring textual similarity. The Embedding Extrema of post(Post-Extrema) is

$$cosine(Extrema(G), Extrema(X))$$

where Extrema is the maximum of the absolute value of each dimension. The Embedding Extrema of reply(Reply-Extrema) is

$$cosine(Extrema(G), Extrema(Y))$$

The Embedding Greedy of Post (Post-Greedy) metric is to find the closest word in the post for each word in the model reply. And then calculate the mean over the cosine similarities for each pair. The Embedding Greedy of Reply (Reply-Greedy) metric is to find the closest word in the ground-truth reply for each word in the model reply. Although these metrics do not strongly correlate with human judgements of generated responses[7], we interpret them as measuring topic similarity. The two setting of metric-based evaluation results are shown in Table 2. Results show that the BL model has the best generations in metric-based evaluation. The distinct-bigram of BL model is 0.0587, which improves 15.5% compared with Seq2Seq_att. And the distinct-bigram of MP model is 0.056, which improves 10.2% compared with Seq2Seq_att. In a word, the BL constraint model has the best evaluation measure. All the constrained models has better results than Seq2Seq_att.

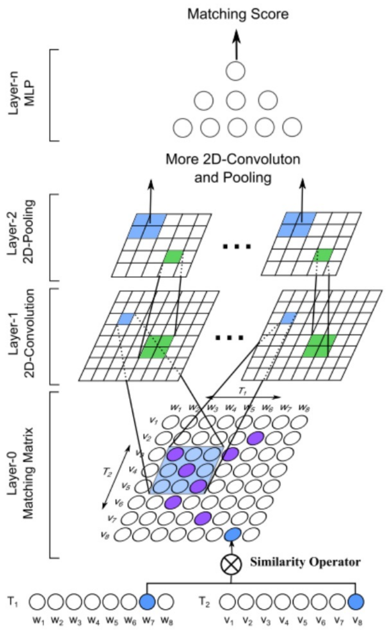

**Figure 1: : An overview of MatchPyramid on Text Matching.**

# 3. RETRIEVAL-BASED METHOD

## 3.1 Interaction-focused model

We take use of MatchPyramid[8] as our model in this part. The main idea comes from modeling text matching as image recognition, by taking the matching matrix as an image, as illustrated in Figure 3.

We start with changing two 1D text representations of words within them to a typically 2D grid.To address this issue, we represent the input of text matching as a matching matrix M, with each element $M_{ij}$ standing for the basic interaction, i.e. similarity between word $w_i$ and $v_j$ calculated by $M_{ij} = w_i \otimes v_j$,$w_i$ and $v_j$ denotes the i-th and j-th word in two texts respectively, and $\otimes$ stands for a general operator to obtain the similarity. Specifically, here we use word2vector to get the vector of each word representation, each query and each comment can be represented by a sequence of word vector, then we apply cosine similarity between each word in the given query and the comment to be evaluated to obtain the matching matrix. The body of MatchPyramid is a typical convolutional neural network, which can extract different levels of matching patterns. It uses the matching matrix mentioned below as input. For the first layer of CNN, the k-th kernel $w^{(1,k)}$ scans over the whole matching matrix $z^{(0)} = M$ to generate a feature map

[1] We used word2vec to train the embedding. The negative sample is 3, the iteration is 20, the embedding size is 300 and the learning rate is 0.05.

**Table 1: The metric-based evaluation results generated from different models on STC.**

| model | dist_uni | dist_bi | Post-Average | Post-Greedy | Post-Extrema | Reply-Average | Reply-Greedy | Reply-Extrema |
|---|---|---|---|---|---|---|---|---|
| Seq2Seq_att | 0.004307 | 0.05082 | 0.4993 | 0.3672 | 0.3032 | 0.5148 | 0.2815 | 0.3010 |
| SIM | 0.00424 | 0.05513 | **0.5167** | **0.3721** | 0.3202 | **0.5323** | 0.2799 | 0.3088 |
| MP | 0.004294 | 0.05593 | 0.5034 | 0.3643 | 0.3088 | 0.5267 | 0.2776 | **0.3097** |
| BL | **0.00438** | **0.0587** | 0.5068 | 0.3148 | **0.3698** | 0.5322 | **0.3048** | 0.2877 |

**Table 2: The MAP results for different models on STC2.**

| model | MAP |
|---|---|
| random | 0.200 |
| BM25 | 0.234 |
| MatchPyramid | 0.484 |

$z^{(1,k)}$:

$$z_{i,j}^{(1,k)} = \sigma\left(\sum_{s=0}^{r_k-1}\sum_{t=0}^{r_k-1} w_{s,t}^{(1,k)} \cdot z_{i+s,j+t}^{(0)} + b^{(1,k)}\right) \qquad (11)$$

where $r_k$ denotes the size of the k-th kernel. Here ReLU is adopted as the active function $\sigma$. After this, a max-pooling is used to get a fixed length pattern vector. We use a use two-layer DNN to produce the final matching score.

$$s = W_2\sigma(W_1 z + b_1) + b_2 \qquad (12)$$

After calculating the query with all comments in the corpus, we use the score to sort the comments and return the top-k of them as the retrieval results.

## 3.2 Experiments

We use python API jieba to perform tokenization and then discard the stop words. After that, we use word2vec to get word representation vectors with dim 50. We use two different size of kernels in CNN and batch size as 200. All the models share the word embeddings trained on the STC2 training data for STC2 Dataset. We adopt the Adam method with learning rate as 0.1. We use BM25 as baseline and the results are shown in table 3. Besides, we show the random results as random candidate ranking.

As we can see from the results, the BM25's result is only slightly better than the random one. We think the reason is that in this task, the query and the comment which share common words are likely to be unrelated or have low score. So BM25 is less likely to achieve good results in this task as it always do in traditional retrieval tasks. Our model tries to find the underlying relationship between the query and the comment, which results in a better performance than the BM25 method.

# 4. CONCLUSION AND FUTURE WORK

This paper reports generation-based method and retrieval-based method in NTCIR-13 STC2 Chinese task. As for generation-based methods, we proposed a constraint of post and reply to the objective function, in order to solve the problem that traditional Seq2Seq model is prone to generate safe and common reply, e.g. I don’ t know. Our results demonstrate that the proposed constrained models gain a boost on topic similarity and response diversity. As for retrieval-based methods, our method chooses the comment which is similar with the given query sharing same words. In the future, we will devote to exploit the topic information for the generation process, which can further improve the diversity of response generation.

# 5. ACKNOWLEDGEMENTS

This work was funded by the 973 National Basic Research Program of China under grant number 2014CB340401 and National Natural Science Foundation of China under grant number 61232010, 61433014, 61425016, 61472401, 61203298.

# 6. REFERENCES

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

[2] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[3] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *Advances in neural information processing systems*, pages 507–513, 1998.

[4] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[5] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

[6] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

[7] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

[8] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng. Text matching as image recognition. In *AAAI*, pages 2793–2799, 2016.

[9] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301, 2017.

[10] 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.

[11] X. Shen, H. Su, Y. Li, W. Li, S. Niu, Y. Zhao, A. Aizawa, and G. Long. A conditional variational framework for dialog generation. *arXiv preprint arXiv:1705.00316*, 2017.

[12] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In C. J. C. Burges, L. Bottou,

M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 926–934. Curran Associates, Inc., 2013.

[13] 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.

[14] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.