An Ensemble Approach to Conversation Generation

Yimeng Zhuang[1], Xianliang Wang[1], Han Zhang[2], Jinghui Xie[1], and Xuan Zhu[1]

[1]Samsung R&D Institute (SRC-BJ), Beijing, China
[2]School of Software & Microelectronics, Peking University, China
{ym.zhuang, xl0126.wang, jh.xie, xuan.zhu}@samsung.com,
zhanghanss@pku.edu.cn

Abstract. As an important step of human-computer interaction, conversion generation has attracted much attention and has a rising tendency in recent years. This paper gives a detailed description about an ensemble system for short text conversation generation. The proposed system consists of four subsystems, a quick response candidates selecting module, an information retrieval system, a generation-based system and an ensemble module. An advantage of this system is that multiple versions of generated responses are taken into account resulting a more reliable output. In the NLPCC 2017 shared task "Emotional Conversation Generation Challenge", the ensemble system generates appropriate responses for Chinese SNS posts and ranks at the top of participant list.

1 Introduction

Dialogue system plays an important role in human life, and its application field is very extensive, such as train routing [1], intelligent tutoring [2]. Contrary to domain-specific dialog system, open-domain tasks are more fascinating and challenging, since it requires the system to adapt to more diverse user needs. Therefore, traditional rule-based [3] or template-based [4] approaches may be not enough. Nowadays, two promising methods are retrieval-based system [5] [6] and generation-based system [7] [8].

Retrieval system deals with user's query by searching for a most related utterance in a database. This data-oriented method highly depends on the coverage of a database, if the ideal response does not exist in the database, the returned response may seriously degrade user experience. On the other hand, generation-based system always gives brand new response by synthesizing utterances. A weakness of generation-based system is that the generated response tends to be short, universal and meaningless, and sometimes has incorrect grammar. In [9], a novel ensemble of retrieval-based and generation-based dialog system is proposed and achieves good performance.

The objective of this paper is to give a detailed description about our submitted conversation generation system in the NLPCC 2017 shared task "Emotional Conversation Generation Challenge". Our system ensembles the retrieval system's results and a seq2seq generation system's results, and achieves promising performance. Section 2 details on our system's overall architecture, as well as the works on data processing. Section 3 gives the evaluation results for the system, and various aspects of the system are analyzed by case study. Section 4 concludes the whole paper.

2 System Architecture

In the NLPCC Emotional Conversation Generation Challenge [10], the participants are asked to build an one-round dialogue system to generate response in natural language, given a Chinese post and the target user-specified emotion category. The possible emotions include anger, disgust, happiness, like, sadness and other.

2.1 Overview

Due to the posts are sentences collected from Weibo [1], in most cases the length of the post is not exceed 20 words, the task can be seem as a short text conversation problem enhanced by the user-specified emotion category requirement.

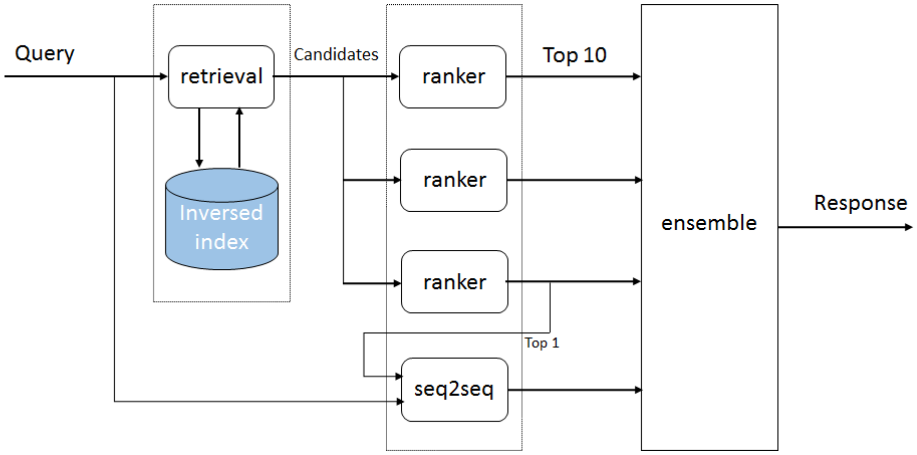

Fig. 1. The overall framework of the proposed method.

Figure 1 indicates the overall framework of our proposed short text conversation generation system. The system are mainly composed of four steps: a quick response candidates selecting module, an information retrieval system, a generation system and an ensemble module.

[1] http://weibo.com

- At beginning, for a given query post q, there are tens of thousands possible post-response pairs $< p, r >$ in the database but only a few post-response pairs have related semantics with the query q. The computational complexity is unacceptable high, if we compare query q to each post-response pair $< p, r >$. The function of the quick response candidates selecting module is to retrieve coarse-grained response candidates efficiently, so as to perform the following complex semantic comparisons only within a small number of candidates.
- Multi-step learn-to-rank neural network models are used to rank the candidates by semantic similarity between the query q and candidate r. Only the top N candidates are remained as the candidate replies. Since there are multiple ranker models, we can obtain different versions of ranking results about the same candidate set that makes the following ensemble result more reliable.
- Seq2seq model based natural language generation is a hot academic topic in recent years, though this technique is still not mature enough as business applications. In our system, we try to implement a seq2seq system as an optional component, which takes the query q and a retrieved candidate response r as input and generates a new utterance as another response candidate.
- Given the response candidates produced by the retrieval and generation approaches, ensemble module re-ranks those responses and outputs the top 1 candidate as the final reply.

2.2 Data Preprocessing

The training data and the testing data provided by the NLPCC 2017 are the only data that we used in our system. The training data set consists of about 1.1 million SNS post-response pairs crawled from Weibo, we split it into three parts, a training set for tuning model, a development set for crossing validation, and a small development set for training ensemble module.

After an arbitrary split, an imperfect ranking model described in section 2.5 can be trained on this data set, and the original training data will be evaluated using this model for data filter. The semantics similarity between each post-response pair is calculated through model, and filter out those post-response pairs that either get score lower than a threshold or the length of any sentence in the post-response pair is less than 3 words. The split result is depicted in table 1.

For the test data, we just remove some insignificant words from the posts, such as '转帖'(repost), '网友制作'(net friend making), etc. As well as, convert full-width alphabet to half-width alphabet.

2.3 Candidates Selecting

The candidates selecting module is accomplished by an inverse indexing. Here, the data is split into six conversation classes according to the emotion label of

Table 1. Data statistics. Small dev set will be filtered manually as described in section 2.7.

-	First Split	Second Split
Training Set	1100000	259150
Dev Set	10000	2049
Small Dev Set	9207	-

the response r in conversation $< p, r >$. For each conversation class, a mapping between words and conversations is built, that is, if a word w exists in either the post p or the response r of a conversation $c =< p, r >$ then word w and conversation c are connected. The mapping is many-to-many, which means different words may correspond to diverse number of conversations, and the number of connected conversations reflects the discrimination capacity of a word. Like the Inverse Document Frequency (IDF), the less the number is, the more unique a word is.

Candidates are selected as following. First, the words in a query post q are sorted in ascending order by the number of connected conversations in the target emotion corresponding inverse indexing. Second, the responses of each word's connected conversations are selected as candidates in the sorted word order until N_{inv} candidates are obtained. In this paper, $N_{inv} = 1000$ is adopted.

In practice, the above co-occurred keyword based approach sometimes may miss appropriate responses particularly when target response emotion is required. Therefore, to relieve this problem, a fixed number of special responses are selected artificially from each emotion class and added as candidates as a supplement. Those special responses have clear sentiment and are general to answer most queries.

2.4 Embedding Pre-training

Word embeddings are obtained by unsupervised learning algorithm for constructing vector representations for words. In a good embedding space, word embeddings map semantic meaning into a geometric space and the distance between any two vectors captures the semantic relationship between that two associated words.

In order to achieve a better training result, the lexicon is well-designed in the proposed system. Instead of using word-level embedding or character-level embedding simply, high-frequency words and Chinese characters together constitute the lexicon. In the system, two sets of lexicon are generated for training different ranking models, one contains 3532 entities formed by words appeared more than 2000 times in the training data and Chinese characters with frequency larger than 120 and all other characters are forced to be mapped to UNK. Another set of lexicon contains 10179 entities consisted of high-frequency words appeared more than 100 times in the training data and Chinese characters with frequency larger than 120 and an UNK label corresponding to other low-frequency characters. The motivation is straightforward, but in practice this is an effective measure for training better models.

Word embeddings are computed via word2vec toolkit [11], which applies a shadow neural network to discover the co-occurence statistics between words in a corpus of text. The detailed configurations and parameters are shown as follows,

```
./word2vec -train train.data -output skipgram.txt -size 100 -window 8
-sample 1e-4 -negative 5 -hs 1 -binary 0 -cbow 0 -iter 15 -min-count 1
-nthread 12
```

Here, the file ‘train.data’ is a preprocessed training corpus, in which words and characters have been mapped into lexicon entities as mentioned previously.

2.5 Learn-to-rank Model

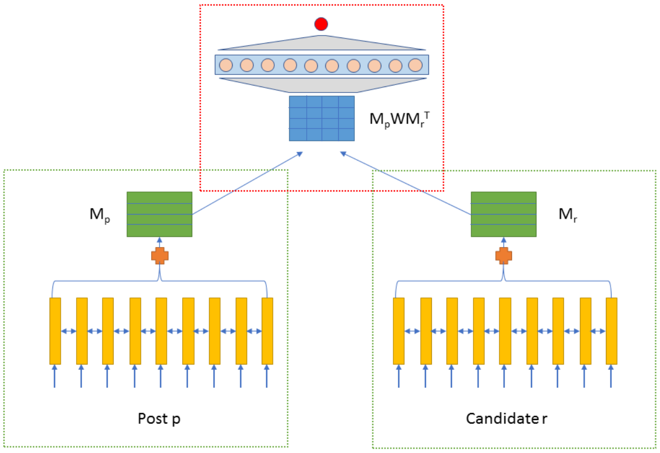

Fig. 2. The structure of the ranking model.

Model Structure In this paper, learn-to-rank models are used to rank those candidates by the semantic similarity. The overall structure of our model for semantics ranking is depicted in Figure 2.

The model consists of two part, one is the sentence embedding extractor corresponding to the green dashed rectangle part in the figure, another is the semantics similarity computing structure corresponding to the red dashed rectangle part in the figure.

In the sentence embedding extractor, we use same bidirectional LSTM and attention mechanism for the input query and response candidate, which means the parameters of this part model are shared. The input of this part model are the word embeddings, which are pre-trained as described in section 2.4 and fixed in the training process. Here, a structured self-attentive sentence embedding [12] is adopted. In this approach, sentence's semantics is represented by a 2-dimensional matrix rather than the widely used vector representation, an advantage of this approach is that the embedding matrix attends on different parts of the sentence. Formally, the weight matrix A produced by the attention mechanism is,

$$A = softmax(W_{s2}tanh(W_{s1}H^T)) \tag{1}$$

Where H is the biLSTM hidden states, W_{s1} and W_{s2} are two trainable parameter matrices. The row number of W_{s2} reflects the number of different parts to be extracted from the input sentence. In this paper, the row number of W_{s2} is 20 and the dimension of hidden state vector is 300. Therefore, the resulting sentence embedding matrix is,

$$M = AH \tag{2}$$

In the semantics similarity computing structure, a 2-dimensional bilinear model firstly makes the local decisions on different parts of sentence p and sentence r by

$$match(p, r) = M_p W M_r^T \tag{3}$$

Where M_q and M_r denote the sentence embedding matrix of p and r respectively, W is a parameter matrix. Since each row of the sentence embedding matrix attends a part of a sentence, after matrix multiplications each element of the resulting matrix $match(p, r)$ reflects a local semantics similarity.

The final decision is made considering all the local decisions through a fully connected neural network above the bilinear model. It outputs a score represents the semantics similarity between the sentence p and r.

Training We train the ranking model in two steps. For a conversation $< p, r^+ >$ in the training set, we firstly select 10 other responses from the N_{inv} candidates of the post p as negative samples by random. So there will be 10 times training cases than the original training set. Each training case consists of a post, a correct response and an incorrect response, which can be denote as a triple $< p, r^+, r^- >$. In the first step, the parameters of a ranking model are tuned on those data. After having the first well-tuned ranking model, the candidates of each post in training set are ranked by this model, and re-sample 10 negative samples only from the top 100 candidates of each post. In the second step, another ranking model is trained from scratch on these new training cases. The reason is that randomly selected negative samples may have less semantics relation with the post while the correct response always has a strong semantics similarity with the post, which leads to an easy case for the ranking model to differentiate and makes limited contribution for model learning. By re-sampling negative samples, the ranking model can learning more information from confusion data.

In training process, for a particular mini-batch of training cases, the max-margin loss function is optimized,

$$loss = \frac{1}{N}\sum_{i=1}^{N} max(0, margin - (score_i^+ - score_i^-)) + P \tag{4}$$

Where N is the size of a mini-batch, $score_i^+$ and $scpre_i^-$ are represent the semantics similarities of the correct response and the negative sample respectively, $margin$ is a hyper-parameter. In our paper, $N = 256$ and $margin = 0.10$. P is the penalization term for self-attention mechanism proposed in [12].

In order to do an more elaborate re-sampling and data filter, we add a regular term on the loss function when training the first ranking model to make the scores concentrate around zero.

$$R = \frac{1}{N}\sum_{i=1}^{N} \beta \left| score_i^+ \right| \tag{5}$$

Where β is a hyper-parameter.

2.6 Generation-based Method

In recent years, there has been a rising tendency to the research of the generation-based method. The generation-based method usually builds an end-to end trainable system using neural networks and it can generate variable utterances.

The method used in the system is based on the "bidirectional sequence to sequence with attention" (biseq2seq-attention) model and it is like an ensemble of retrieval- and generation-based method [9] in some ways.

The overall architecture of the generation-based method is depicted in Figure 3. Two contributive mechanisms are integrated in the architecture: (1) In the encoding phase, the retrieve model is ensembled to get retrieved query. And the embeddings of the original query and retrieved query are concatenated; (2) After the decoding, Diverse Beam Search algorithm [13] is adopted to decode a list of diverse outputs.

Given an user query sequential object $X = [x_1, ..., x_{Lx}]$, the vocabulary is embedded by looking up the pre-trained embedding table which is trained using word2vec tool [11]. Then an encoding GRU transforms the vector sequence into an encoded representation E_1. In the meanwhile, retrieval-based system is utilized to retrieve the analogous query sequential object $Y = [y_1, ..., y_{Ly}]$ from the data base. The retrieved query is also encoded into a retrieved representation E_2. The two vectors are concatenated, and an decoder GRU is modelled to generate the target sequence $O = [o_1, ..., o_{Lo}]$.

In the results, the dimension of the word embedding was 100. The utterances with out-of-vocabulary are removed in the training to get better models. Single-layer GRU was used and the dimension of the hidden layer was 220.

After the decoding, Diverse Beam Search algorithm is used to decode a list of diverse output by optimizing for a diversity-augmented objective [13]. It divides

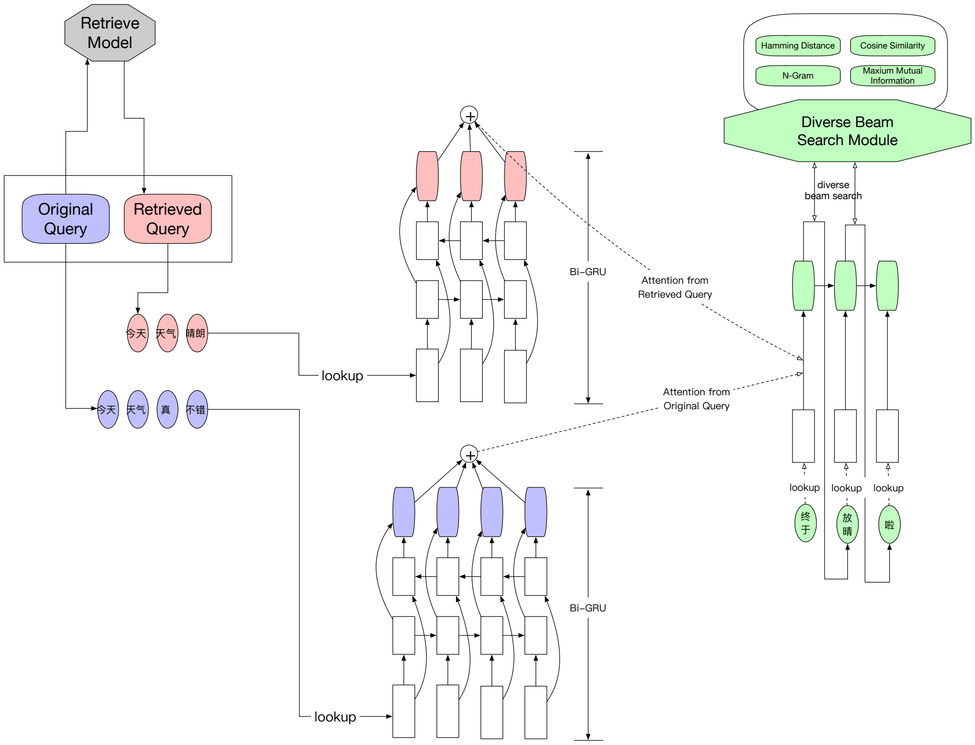

Fig. 3. The overall architecture of the generation-based method

the beam budget into groups and enforcing diversity between groups of beams. The Diverse Beam Search is a doubly greedy approximate inference algorithm which greedily optimizes the diversity-augmented model score along both time and groups. Results show the algorithm can produce more diverse reply than the traditional Beam Search Algorithm.

The ranker is them used to select the most matched reply from the diverse list. Hamming Distance, Cosine Similarity, N-Gram and Maximum Mutual Information are used in the ranker.

2.7 Ensemble

In the system, a set of 2-step ranking models, a set of 3-step ranking models, and a seq2seq model are trained, therefore, for a query there will be at most 60 unique responses generated from those six models. The small dev set mentioned in section 2.2 is used for tuning the ensemble model. For each post in the small dev set, 20 to 60 unique responses are firstly generated by the six models, and then these responses are labeled as suitable or unsuitable manually. We finish this by crowd-sourcing, and 1922 posts are labeled in total. These labeled cases constitute the training set for ensemble model.

The ensemble model used in this system is a linear ranking model based on xgboost [14] with pairwise ranking objective using linear booster. The input features for ranking responses include five semantics similarity scores from the five ranking models, the emotion labels, the source (generated by which model), the length of sentence, a language model score, and the five simply matching features proposed in [15]. When training ensemble model using xgboost, each post's responses form a group and responses labeled as suitable rank in front of responses labeled as unsuitable. In runtime, given a query and its responses produced by ranking and seq2seq models, extract input features firstly, and then the ensemble model ranks these responses and returns only the top 1 response as the system's final result.

3 Experiments

3.1 Experimental Setup

The test data contains more than 5000 Weibo posts, participants are required to generate one response per emotion for each post. But due to it is a huge effort for manually assessing, only 200 posts are selected and checked by human. Which post will be selected is unknown for the participants for fair comparison. It is guaranteed that the selected data has clear emotion and is fluent.

The evaluation for submitted responses considers three metrics including content coherence, fluency and emotion consistency. If a response is appropriate in terms of both logic and content and is fluent in grammar, it can get 1 point and will check whether the emotion of this response is the same as the pre-specified emotion, if the emotion is consistent too, then the response will get another point. The final performance is evaluated by the sum and average score of all the test cases. Manual assessing is finished by voting among three evaluators.

Table 2. Overall score and the performance of each emotion class.

Submission/Emotion	0 point	1 point	2 points	Count	Total score	Average score
samsung_1_EGG	267	82	247	596	576	0.966
samsung_2_EGG	272	93	231	596	555	0.961
Like	50	16	67	133	150	1.128
Sad	56	15	48	119	111	0.933
Disgust	88	24	32	144	88	0.611
Angry	37	5	29	71	63	0.887
Happy	36	22	71	129	164	1.271

3.2 Results

Table 2 shows the final performance of our systems, samsung_1_EGG and samsung_2_EGG are our two submissions whose only difference is the lexicon, samsung_1_EGG has 3532 entities while samsung_2_EGG has 10179 entities as described in section 2.4.

As shown in the table, the ensemble approach achieves fairly good performance. samsung_1_EGG is slightly better than samsung_2_EGG indicates that introducing appropriate amount of high-frequency words into lexicon may help improve the overall performance. The last five lines give the statistics of samsung_1_EGG on every emotion. It is clear that the performances on positive emotions are much better than negative emotions.

Table 3. An randomly selected example of generated responses.

Query	你要狠狠喜欢自己，才会有人很很喜欢你!
Emotion	Generated responses
Like	好吧，我也是，我喜欢你，我也喜欢你喜欢的人
	真的很喜欢你，我喜欢的人，我也会喜欢的，我也爱你
	其实我也会觉得很好的人，我也爱你，我也会喜欢的
	爱你的人都会爱我的，我也会爱你的。
	也会很好。但人也很好的，我也会喜欢的人。
Disgust	是啊，我也不知道你是男人，你也不喜欢你，我也不喜欢你
	其实我也不知道你是男人，你也不喜欢你的人。
	说得对！我也不喜欢你，你也不知道你的意思!
	没人说的，我也不知道你是男人，你也不知道你的意思。
	我们都是这样的，但我也不喜欢你的人。你也不知道你的意思。

3.3 Case Study of the seq2seq results

Case study is an intuitive way to get a better understanding of the effectiveness of the proposed system. Table 3 illustrates the results of the seq2seq model through several examples. The examples show that the generated responses tend to be general, and include many repeated patterns. For example, in this table, " 也喜欢"(also like) appears so many times, though it does not affect expression, it makes the sentence weird.

Table 4. The effectiveness of ensemble module.

Query	最大的快乐莫过于平凡.			
Emotion	Labels	Candidate responses	Ranker	Ensemble
Sadness	Suitable	可是我不快乐	14	1
	Suitable	不快乐，我悲催	2	2
	Suitable	不快乐，日子太无聊了	5	3
	Suitable	原本快乐多过痛苦，现在痛苦多过欢乐	24	4
	Suitable	那就不快乐了，一起痛苦吧。	11	5
Happiness	Suitable	的确。最开心的是莫过于此	1	1
	Suitable	最开心的事莫过于此.	2	2
	Suitable	哈哈因为我们很平凡	28	3
	Suitable	最开心的莫过于你们说的满意俩字了!	12	4
	Unsuitable	高一是最开心的!	3	5

3.4 Analysis of the ensemble module

The ensemble module is one of the most important components in our system, table 4 illustrates the changes after the introduction of ensemble module. From the table, we can clearly see that more appropriate responses are ranked higher by the ensemble module, while unsuitable responses are degraded. More specifically, the sentence ” 可是我不快乐”(But I'm not happy.) looks more like a response replied by human, because of the transitional word ” 可是”(but).

4 Conclusions

In this paper, we give a detailed description about an ensemble system for short text conversation generation. The system filters out most unrelated utterances by a quick candidates selecting module, and then several rankers take the candidates as input and output the top 10 best responses. The system also uses a generation-based method as a supplement to increase the diversity of response. At last, responses produced by rankers and generation are ranked by the ensemble module, and system returns final response. Although the experiments are conducted on an emotion conversation generation task, the results is fairly good. Besides, a few case studies are conducted in this paper.

References

[1] Sikorski, Teresa, and James F. Allen. ”A task-based evaluation of the TRAINS-95 dialogue system.” Workshop on Dialogue Processing in Spoken Language Systems. Springer, Berlin, Heidelberg, 1996.

[2] Litman, Diane J., and Scott Silliman. ”ITSPOKE: An intelligent tutoring spoken dialogue system.” Demonstration papers at HLT-NAACL 2004. Association for Computational Linguistics, 2004.

[3] Clancey, William J. ”Tutoring rules for guiding a case method dialogue.” International Journal of Man-Machine Studies 11.1 (1979): 25-49.

[4] Levin, Esther, Roberto Pieraccini, and Wieland Eckert. ”Using Markov decision process for learning dialogue strategies.” Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on. Vol. 1. IEEE, 1998.

[5] Huang, Chao, et al. ”LODESTAR: A Mandarin Spoken Dialogue System for Travel Information Retrieval.” Sixth European Conference on Speech Communication and Technology. 1999.

[6] Eric, Mihail, and Christopher D. Manning. ”Key-Value Retrieval Networks for Task-Oriented Dialogue.” arXiv preprint arXiv:1705.05414 (2017).

[7] Serban, Iulian Vlad, et al. ”Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models.” AAAI. 2016.

[8] Li, Jiwei, et al. ”Deep reinforcement learning for dialogue generation.” arXiv preprint arXiv:1606.01541 (2016).

[9] Song Y, Yan R, Li X, et al. Two are Better than One: An Ensemble of Retrieval- and Generation-Based Dialog Systems[J]. 2016.

[10] Hao Zhou, Minlie Huang, Xiaoyan Zhu, Bing Liu. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. arXiv:1704.01074.

[11] http://code.google.com/p/word2vec/

[12] Lin, Zhouhan, et al. ”A structured self-attentive sentence embedding.” arXiv preprint arXiv:1703.03130 (2017).

[13] Vijayakumar A K, Cogswell M, Selvaraju R R, et al: Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models[J]. 2016.

[14] https://github.com/dmlc/xgboost

[15] Ji, Zongcheng, Zhengdong Lu, and Hang Li. ”An information retrieval approach to short text conversation.” arXiv:1408.6988 (2014).