Attention Based Dialogue Context Selection Model

Weidi Xu[1], Yong Ren[2], and Ying Tan[1(✉)]

[1] Key Laboratory of Machine Perception (Ministry of Education) and Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, People's Republic of China
wead_hsu@pku.edu.cn, ytan@pku.edu.cn

[2] Complex Engineered System Lab (CESL), Department of Electronic Engineering, Tsinghua University, Beijing 100084, People's Republic of China
reny@tsinghua.edu.cn

Abstract. The particular phenomena of Information Overload and Conversational Dependency in multi-turn dialogues have brought massive noise for feature learning in existing deep learning models. To solve the problem, the Attention Based Dialogue Context Selection Model (ABDCS) is proposed in this paper. This model uses attention mechanism to extract the relationship between current response utterance and previous utterances. Qualitative and quantitative analysis show that ABDCS is able to choose the semantically related utterances in its dialogue history as context and be robust against the noise.

1 Introduction

Dialogue System [1] is computer system that interacts with humans through natural language. It involves many frontier research directions such as natural language understanding, information retrieval, logical reasoning, text generation and speech recognition. In recent years, thanks to the breakthrough in the NLP field, data-driven dialogue system has become the research hotspot.

Dialogue system, according to the way of interaction, can be divided into single-turn and multi-turn dialogue system. Multi-turn dialogue, which are usually seen in the open-domain scenario, is more liberal and usually consists of interwoven interrogation questions. The multi-turn dialogue systems should take long-term historical information into consideration, which requires sophisticating logical reasoning.

This paper considers two major problems in multi-turn dialogue, i.e., the information overload and conversational dependency. The information overload is very common in the conversations among a group of people, e.g., the barrage of Twitch TV [10]. The users have limited capability of processing information in the conversations. Therefore, when the responses are produced quickly, the users will be overwhelmed by numerous utterances and have to perceive the information selectively. Figure 1 illustrates the curve about the rate of response

L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11302, pp. 286–295, 2018.
https://doi.org/10.1007/978-3-030-04179-3_25

and information bandwidth. When the rate is low (on the left), the information can be easily processed and the replies are generated quickly. On the other side, when the information is too heavy to be received by individuals, the rate will be decreased. Due to this phenomenon, the agent should better process the sentences selectively in the multi-turn dialogue.

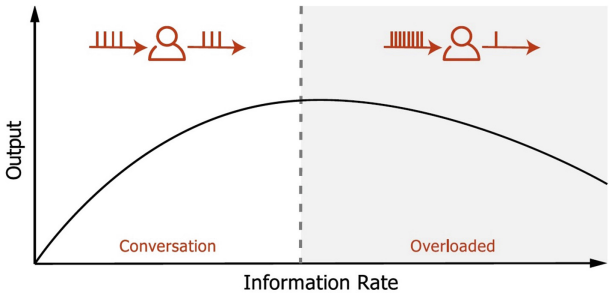

Fig. 1. The illustration of information overload [10].

In terms of conversational dependency, [6] and [5] demonstrates that each utterance has a corresponding previous utterance that is most semantically related. The dependencies of the utterances can be used to parse the dialog structure, as shown in Fig. 2.

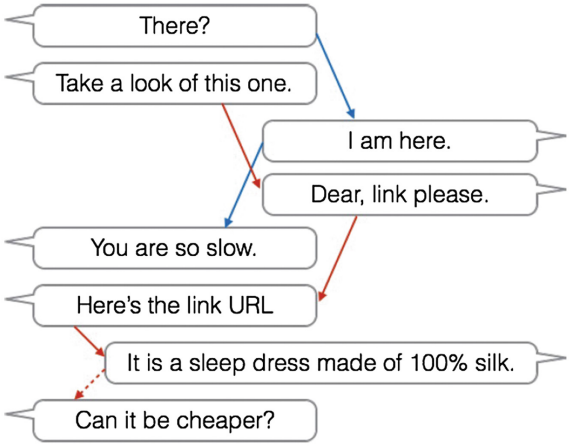

Fig. 2. The dependency of utterances [5]

The study of the dependency of utterance is of great significance. The interdependence between utterance helps us to analyze the structure of the dialogue and

can be applied to other tasks, e.g., dialogue generation, topic analysis. Especially in conversational tasks, the information in previous utterances plays a crucial role in the generation of statements [14]. By learning the dependencies between the sentences, the relevant sentences in the dialogue can be selected more accurately while ignoring irrelevant information and noise, in order to improve the quality of the dialogue.

The research on the interdependence of utterances is still at a relatively preliminary stage. The typical method directly use the predefined rule [13], or simply derive the dependency via discrimination models [5]. The Attention Based Context Selection Model (ABCSD) is proposed, which is able to extract the interdependency between utterance in an unsupervised learning manner. Specifically, the model first select a certain previous utterance, based on which a new reply is generated.

The main contributions of this paper are as follows:

1. The ABDCS model is proposed, which tries to use the attention mechanism to establish the relationship between the utterances. The comparative experiments on the Chinese-English dialogue dataset show that compared with the existing models, this method outperforms.
2. It is the first time to use unsupervised generative model to obtain the dependencies of the conversational statements. Qualitative analysis shows that the dialog dependencies obtained by this method can effectively extract the key information of dialogues and the parallel structure of the dialogue.

2 Model

2.1 Framework

When generating statements in a dialogue, a previous utterance that is most informative for generating current response is selected and it is used as the main context in producing the utterance. When doing this, we use the attention mechanism to automatically extract the dependencies between statements. In this work, the "statement", "utterance" and "response" are used exchangeable. The proposed model, referred as (*Attention Based Dialogue Context Selection Model (ABDCS)*) is shown in Fig. 3.

Briefly, the *ABDCS* model uses a recurrent neural network to encode the dialogue statements independently into the statement vector D (F_{BiLSTM} in the figure), and simultaneously use the LSTM language model for dialog generation (F_{LM} in the figure). Similar to *HRED* [12], *ABDCS* also uses the dialog-level recurrent neural network (F_{RNN} in the figure) to describe the dialogue state R. The entire model is shown by the Algorithm 1.

Sentence Encoding via Bidirectional Recurrent Neural Networks. We use bidirectional LSTM network [3,4,11] to encode the raw input sentences, including a forward LSTM networks F_{LSTM}^{fw} and a backward LSTM networks F_{LSTM}^{bw}.

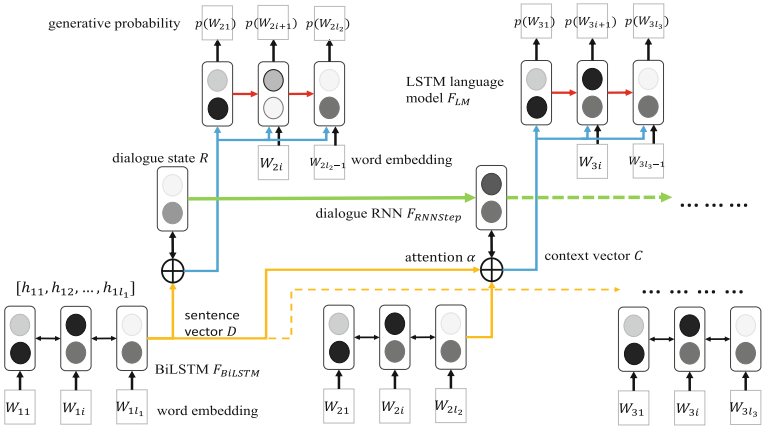

Fig. 3. The framework of the *ABDCS*.

Algorithm 1. *ABDCS* Model

Require: Dialgoue set $D =< S_1, S_2, \cdots, S_{l_D} >$
Ensure: Loss J

1: Initializes $J = 0$, $t = 1$, and dialog state $R = \mathbf{0}$, $D_0 = \mathbf{0}$;;
2: Encoding dialog into vectors $< D_0, D_1, D_2, \cdots, D_{l_D} > \xleftarrow{\text{word vector}}{\text{RNN}} < S_1, S_2, \cdots, S_{l_D} >$;
3: **repeat**
4: Calculating attention $\alpha \xleftarrow{\text{Attention mechanism}} (R, < D_0, D_1, \cdots, D_{t-1} >)$;
5: Calculates the above vector $C \xleftarrow{\text{weighted sum}} (\alpha, < D_0, D_1, \cdots, D_{t-1} >)$;
6: Calculates the loss of S_t $J'_{S_t} \xleftarrow[\text{language model}]{\text{LSTM}} (C, S_t)$;
7: Update loss $J = J + J'_{S_t}$;
8: Updates the dialog state $R \xleftarrow[\text{status update}]{\text{RNN}} (R, C, D_t)$,$t = t + 1$;
9: **until** $t > l_D$

Let $W_{i,j}$ denote j-th word in the sentence S_i, $h^{fw}_{i,j}$ and $h^{bw}_{i,j}$ be the state of forward LSTM and backward LSTM respectively, the vector D_i presenting the entire sentence can be formulated by:

$$[h^{fw}_{i,1}, h^{fw}_{i,2}, \cdots, h^{fw}_{i,l_{S_i}}] = F^{fw}_{LSTM}(S_i), \ h^{fw}_{i,j} \in \mathbb{R}^{U_{rnn}} \tag{1}$$

$$[h^{bw}_{i,1}, h^{bw}_{i,2}, \cdots, h^{bw}_{i,l_{S_i}}] = F^{bw}_{LSTM}(S_i), \ h^{bw}_{i,j} \in \mathbb{R}^{U_{rnn}} \tag{2}$$

$$D_i = [h^{fw}_{i,l_{S_i}}, h^{bw}_{i,1}], \ V_i \in \mathbb{R}^{U_{rnn} \times 2} \tag{3}$$

After the $S_{<t}$ is processed with the bidirectional LSTM, the vectors of the preceding statements are represented as $< D_1, D_2, \cdots, D_{t-1} >$.

Attention Mechanism. After encoding each sentence into a vector, the dialogue can be processed in the sentence-level semantic space. Denote the attention weight of each sentence in $S_{<t}$ as the vector $\alpha_t = (\alpha_{t,1}, \alpha_{t,2}, \cdots, \alpha_{t,t-1})$, where $\sum_{k=1}^{t-1} \alpha_{t,k} = 1$, and denote the dialog state R_t to describe the state of the entire dialog before generating the first t statements, the dialogue state transforms as:

$$\begin{aligned} R_t &= F_{RNNStep}(R_{t-1}, D_{t-1}, C_{t-1}) \\ &= \tanh(W_h[R_{t-1}, D_{t-1}, C_{t-1}] + b_h) \end{aligned} \tag{4}$$

$$C_t = \sum_{k=1}^{t-1} \alpha_k D_k \,,\ C_t \in \mathbb{R}^{U_{rnn} \times 2} \,, \tag{5}$$

where C_t is the previous vector (*Context Vector*) that the model used to generate the t-th statement. We use the dialog state R_t and the preceding statement's $< D_1, D_2, \cdots, D_{t-1} >$ to calculate the attention weight:

$$\phi_{t,i} = F_{att}(R_t, D_i) \tag{6}$$

$$\alpha_{t,i} = \frac{\exp(\phi_{t,i})}{\sum_{k=1}^{t-1} \exp(\phi_{t,i})} \tag{7}$$

$$F_{att}(R_t, D_i) = \begin{cases} V^T \tanh(W[R_t, D_i] + b) & \textit{concat} \\ R_t^T M D_i & \textit{bilinear}, \end{cases} \tag{8}$$

where $\phi_{t,i}$ is the attention score and F_{att} is the attention score function.

Dialogue Generation. This article uses the LSTM network to generate the dialog statement S_t. The LSTM uses the vector C_t as the condition input as well as the preceding $j-1$ words $w_{t,1}, w_{t,2}, \cdots, w_{t,j-1}$ to predict the probability of generating j-th word $w_{t,j}$. Let the implicit state of the LSTM language model be $g_{t,j-1}$, and the entire process is given by the following conditional probabilities:

$$\begin{aligned} p(w_{t,j}) &= p(w_{t,j}|C_t, w_{t,1}, w_{t,2}, \cdots, w_{t,j-1}) \\ &= p(w_{t,j}|C_t, W_{t,j-1}, g_{t,j-1}) \\ &= p(w_{t,j}|g_{t,j}) = softmax(W_V g_{t,j} + b_v), \end{aligned} \tag{9}$$

where $W_V \in \mathbb{R}^{|V| \times U_{rnn}}$ and $b_V \in \mathbb{R}^{|V|}$ and $|V|$ represents the vocabulary size.

Loss Function. The loss function of the LSTM language model is the log likelihood function of the statement. In the dialog generation task, we optimize the model by minimizing the negative logarithm generation probability of each statement in the conversation.

$$\begin{aligned} J(D) &= \sum_{i=1}^{l_D} -\log p(S_i) \\ &= \sum_{i=1}^{l_D} \sum_{j=1}^{l_{S_i}} -\log p(w_{i,j}|C_i, W_{i,j-1}, G_{i,j-1}) \end{aligned} \tag{10}$$

As every function in the *ABDCS* model is differentiable, the model can be trained by the end-to-end stochastic gradient descent method (*SGD*).

3 Experiments

Deep model requires a large corpus. The experiments in this paper are mainly based on the following data sets:

Ubuntu-Chat [9]. The Ubuntu-Chat dataset is a multi-round dialogue data set which focuses on Ubuntu system-related issues. This article uses the NLTK natural language processing package [2] to preprocess the original corpora in the data set. The training data consists of 1.5 million conversations, the size of validation and test sets is 10,000. We use *Ubuntu* to refer to this dataset in the following.

Ubuntu-Chat-200K. In order to compare the effects of amount of training samples, 200K conversations were randomly selected from the *Ubuntu* data set as training sets. The preprocessing methods were identical to the *Ubuntu* data sets. Similarly, we use *Ubuntu-200K* to refer to this dataset.

Baidu Tieba Data. This article uses crawlers to randomly grab online text from Baidu Tieba and creates an open field Chinese multi-round dialogue data set. We use *Tieba* to refer to this data set. Finally, training data with a data volume of 2 million conversations and validation/test sets with a size of 10,000 were obtained.

Table 1. Statistical summary of *Ubuntu* and *Tieba*

Data sets	Fields	Training sets	Verification/ Test sets	Means rounds	Mean words	Total sentences	Total word counts	Lexword size
Ubuntu	Computers	1.5 Million	1 Million	6.9	About 70	103.780 Million	1.05 Billion	21157
Tieba	Open	2 Million	1 Million	5.3	About 45	1063.3 Million	9047.8 Million	28777

The detailed statistics of the data set are summarized in the Table 1.

3.1 Evaluation Metrics and Results

Perplexity. Perplexity (*Perplexity, PPL*) is a common index for evaluating natural language generation models. Its basic definition is:

$$PPL(S) = \exp\left[\frac{1}{n_S} \cdot \sum_{w_i \in S} -\log p(w_i)\right], \tag{11}$$

where sentence S has n_S words. In the dialogue generation task, in order to describe the generation of the dialogue from different perspectives, we slightly adjust the calculation of the confusion degree and define it in the form of (12) and (13).

$$PPL(D) = \exp\left[\frac{1}{n_D} \cdot \sum_{S_i \in D} \sum_{w_{i,j} \in S_i} -\log p(w_{i,j})\right] \tag{12}$$

$$PPL@L(D) = \exp\left[\frac{1}{n_{S_{|D|}}} \cdot \sum_{w_j \in S_{|D|}} -\log p(W_j)\right], \tag{13}$$

where C represents a complete conversation containing a total of n_D words, and $S_{|D|}$ represents the last statement in conversation D, containing $n_{S_{|D|}}$ words. The metric (12) is used to calculate the perplexity of the entire dialogue, reflecting the overall situation of the dialogue; The formula 13 calculates the perplexity of the last reply. The results are summarized in Tables 2, 3 and 4. ABDCS-WA is the ABDCS model with the word-level attention mechanism.

Table 2. *Ubuntu* dataset perplexity (*PPL*) results on validation sets/test sets

Ubuntu	PPL	PPL@L
LSTM	43.30/44.97	43.15/44.66
HRED	43.82/44.08	44.02/44.81
ABDCS	42.53/43.29	42.74/43.51
ABDCS-WA	**41.79/42.14**	**41.92/42.27**

Table 3. *Ubuntu-200K* dataset perplexity (*PPL*) results on validation sets/test sets

Ubuntu-200K	PPL	PPL@L
LSTM	49.89/51.35	49.12/50.88
HRED	48.53/49.92	48.91/50.57
ABDCS	47.83/49.15	47.85/49.45
ABDCS-WA	**46.40/47.56**	**46.59/47.82**

It should be noted that this article uses the perplexity *PPL* as the only criteria for evaluating the optimal model. We used the early stop strategy [7] in the training to avoid overfitting.

Table 4. *Tieba* dataset perplexity (*PPL*) results on validation set/test set

Model	PPL	PPL@L
LSTM	123.3/123.7	123.8/124.0
HRED	125.1/126.0	126.7/127.2
ABDCS	120.5/120.8	121.2/121.3
ABDCS-WA	**116.6/117.1**	**116.9/117.4**

Table 5. The word error rate (*WER*) of the *Ubuntu* data set on the validation set/test set

Ubuntu	WER %	WER@L %
LSTM	67.85/68.02	67.57/67.91
HRED	66.45/66.79	66.32/66.65
ABDCS	66.04/66.49	65.97/66.31
ABDCS-WA	**65.25/65.71**	**65.22/65.58**

Word Error Rate. The word error rate (*WER*) [8] is also one of the commonly used indicators for evaluating natural language generation models. Its basic definition is as follows:

$$WER(S) = 1 - \frac{n_D}{n_S}, \tag{14}$$

where S is a sentence containing n_S words, and n_D represents the number of correctly predicted words. The WER of entire dialogue and the WER of last reply is defined as (15) and (16).

$$WER(D) = \frac{\sum_{S_i \in D} \sum_{w_{i,j} \in S_i} \#[w_{i,j} \neq \arg max(p(w|w_{i,<j}))]}{n_D} \tag{15}$$

$$WER@L(D) = \frac{\sum_{w_j \in S_{|D|}} \#[w_j \neq \arg max(p(w|w_{<j}))]}{n_{S_{|D|}}} \tag{16}$$

The word error rate results are summarized in Tables 5, 6 and 7.

Table 6. The word error rate (*WER*) of the *Ubuntu-200K* data set on the validation set/test set

Ubuntu-200K	WER %	WER@L %
LSTM	69.58/70.05	69.36/69.87
HRED	68.12/68.22	68.03/68.16
ABDCS	67.21/67.65	67.29/67.83
ABDCS-WA	**66.87/67.24**	**66.95/67.34**

Table 7. *Tieba* dataset word error rate (*WER*) results on validation set/test set

Model	WER %	WER@L %
LSTM	74.89/75.11	75.01/75.59
HRED	75.68/76.42	75.86/76.78
ABDCS	73.54/73.65	73.67/73.83
ABDCS-WA	**72.26/72.61**	**72.42/72.90**

4 Conclusion

Overall, this article presents the attention-based models *ABDCS* compared to the baseline models *LSTM* and *HRED*. There is a clear improvement in the results of perplexity and word error rate. Experimental results prove that the attention mechanism is very effective in dialog generation.

Acknowledgment. This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61673025 and 61375119 and Supported by Beijing Natural Science Foundation (4162029), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

References

1. Arora, S., Batra, K., Singh, S.: Dialogue system: a brief review. CoRR abs/1306.4134 (2013). http://arxiv.org/abs/1306.4134
2. Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Association for Computational Linguistics, Philadelphia (2002)
3. Cheng, L., et al.: Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks. IEEE Trans. Neural Netw. **22**(5), 714–726 (2011)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid information services for distributed resource sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing. Proceedings, pp. 181–194. IEEE (2001)

5. Du, W., Poupart, P., Xu, W.: Discovering conversational dependencies between messages in dialogs. CoRR abs/1612.02801 (2016). http://arxiv.org/abs/1612.02801
6. Elsner, M., Charniak, E.: Disentangling chat with local coherence models. In: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, Portland, Oregon, USA, 19–24 June 2011, pp. 1179–1189 (2011). http://www.aclweb.org/anthology/P11-1118
7. Hansen, L.K., Larsen, J., Fog, T.: Early stop criterion from the bootstrap ensemble. In: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1997, Munich, Germany, 21–24 April 1997, pp. 3205–3208 (1997). https://doi.org/10.1109/ICASSP.1997.595474
8. Klakow, D., Peters, J.: Testing the correlation of word error rate and perplexity. Speech Commun. **38**(1–2), 19–28 (2002). https://doi.org/10.1016/S0167-6393(01)00041-3
9. Lowe, R., Pow, N., Serban, I., Pineau, J.: The Ubuntu dialogue corpus: a large dataset for research in unstructured multi-turn dialogue systems. CoRR abs/1506.08909 (2015). http://arxiv.org/abs/1506.08909
10. Nematzadeh, A., Ciampaglia, G.L., Ahn, Y., Flammini, A.: Information overload in group communication: from conversation to cacophony in the twitch chat. CoRR abs/1610.06497 (2016). http://arxiv.org/abs/1610.06497
11. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997). https://doi.org/10.1109/78.650093
12. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A.C., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA, 12–17 February 2016, pp. 3776–3784 (2016). http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11957
13. Shen, D., Yang, Q., Sun, J., Chen, Z.: Thread detection in dynamic text message streams. In: SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, 6–11 August 2006, pp. 35–42 (2006). http://doi.acm.org/10.1145/1148170.1148180
14. Sordoni, A., et al.: A neural network approach to context-sensitive generation of conversational responses. In: NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31–June 5 2015, pp. 196–205 (2015). http://aclweb.org/anthology/N/N15/N15-1020.pdf