When Less Is More: Using Less Context Information to Generate Better Utterances in Group Conversations

Haisong Zhang[1], Zhangming Chan[2], Yan Song[1], Dongyan Zhao[2], and Rui Yan[2(✉)]

[1] Tencent AI Lab, Beijing, China
{hansonzhang,clksong}@tencent.com
[2] Institute of Computer Science and Technology, Peking University, Beijing, China
{chanzhangming,zhaody,ruiyan}@pku.edu.cn

Abstract. Previous research on dialogue systems generally focuses on the conversation between two participants. Yet, group conversations which involve more than two participants within one session bring up a more complicated situation. The scenario is real such as meetings or online chatting rooms. Learning to converse in groups is challenging due to different interaction patterns among users when they exchange messages with each other. Group conversations are structure-aware while the structure results from different interactions among different users. In this paper, we have an interesting observation that fewer contexts can lead to better performance by tackling the structure of group conversations. We conduct experiments on the public Ubuntu Multi-Party Conversation Corpus and the experiment results demonstrate that our model outperforms baselines.

Keywords: Group conversations · Context modeling
Dialogue system

1 Introduction

Dialogue systems such as chatbots and virtual assistants have been attracting great attention nowadays [17,18,21,22,27,28]. To launch dialogue systems with moderate intelligence, the first priority for computers is to learn how to converse by imitating human-to-human conversations. Researchers have paid great efforts on learning to converse between two participants, either single-turn [5,14,16,31] or multi-turn [12,25,29,33]. The research is valuable but is still quite simple in reality: two-party conversations do not cover all possible conversation scenarios.

A more general scenario is that conversations may have more than two interlocutors conversing with each other [9,32], known as "group conversations". In real-world scenarios, group conversations are rather common, such as dialogues

H. Zhang and Z. Chan—contribute equally.

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 76–84, 2018.
https://doi.org/10.1007/978-3-319-99495-6_7

in online chatting rooms, discussions in forums, and debates, etc. Learning for group conversations is of great importance, and is more complicated than two-party conversations which requires extra work such as understanding the relations among utterances and users throughout the conversation.

Table 1. An example of group conversations in the IRC dataset. The conversation involves multiple participants and lasts for multiple turns.

User	Utterance
User 1	"i'm on 15.10"
User 2	@User 1 "have you tried using... "
User 1	@User 2 "nope. but this might... "
User 3	"i read on the internets..."
User 2	"yeah. i'm thinking ..."

For example, in Ubuntu Internet Relay Chat channel (IRC), one initiates a discussion about an Ubuntu technical issue as illustrated in Table 1: multiple users interact with each other as a group conversation. Different pieces of information are organized into a structure-aware conversation session due to different responding relation among users. Some utterances are closely related because they are along the same discussion *thread*, while others are not. We characterize such an insight into the structure formulation for group conversations.

Group conversations are naturally multi-party and multi-turn dialogues. Compared with two-party conversations in Fig. 1(a), a unique issue for group conversations is to incorporate multiple threads of interactions (which indicate "structures") into the conversations formulation, as illustrated in Fig. 1(b).

Learning to generate utterances in group conversations is challenging: we need a uniform framework to model the structure-aware conversation sessions, which is non-trivial. To this end, we propose a tree-structured conversation model to formulate group conversations. The branches of the tree characterize different interaction threads among the users. We learn to encode the group conversation along the tree branches by splitting the tree as sequences (Fig. 1(c)). Given the learned representations, the model generates the next utterance in group conversations. Due to the tree-structured frame of group conversations, we will not use all utterances across turns to generate the target utterance: only the utterances along the target tree branch will be used. With **fewer** contexts used, we obtain even **better** generation results. It is interesting to see that "less" is "more".

To sum up, our contributions in this paper are:

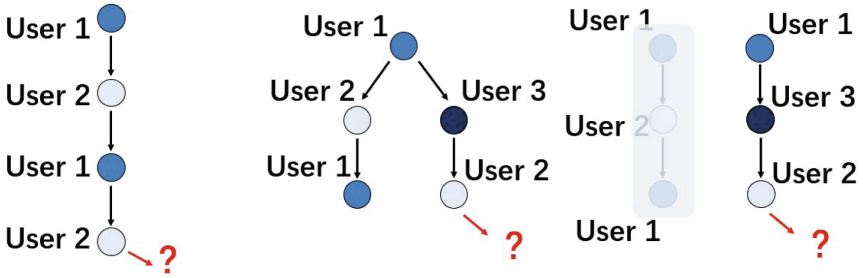
- We are the first to investigate structure-aware formulation for group conversations. The model organizes the utterance flows in the conversation into a tree-based frame, which is designed especially for the group conversation scenario.

- To the best of our knowledge, we are also the first to investigate the task of generation-based conversation in groups. With less context information used by ruling out utterances from irrelevant branches, we generate better results.

Experiments are conducted on the Ubuntu Multi-party Conversation Corpus, a public dataset for group conversation studies. The experimental result shows that our model outperforms the state-of-the-art baselines on all metrics.

2 Background Knowledge

Dialog systems can be categorized as generation-based and retrieval-based approaches. Generation-based methods produce responses with natural language generators, which are learned from conversation data [5,14,17,23]; while retrieval-based ones retrieve responses by ranking and selecting existing candidates from a massive data repository [24,25,27,30]. Researchers also investigate how to ensemble generation-based ones and retrieval-based ones together [11,15]. In this paper, we focus on the generation-based conversational model.



(a) Two parties. (b) Tree structure. (c) Tree splitting.

Fig. 1. We illustrate the difference of conversations between two participants in (a) and the group conversation of Table 1 in (b). Group conversations are structure-aware and formulated as trees (b) and we split the tree into sequences (c). "Irrelevant" utterances on other sequences are not used for generation (shaded in the figure).

Early work in dialog systems focuses on single-turn conversations. It outputs a response given only one utterance as the input [5,14,20]. However, a normal conversation lasts for several turns. Conversation models without context information is insufficient [22]. Context representation learning is proposed to solve the problem. Interested readers may refer to a recent survey paper for more details [26].

Multi-turn dialogue systems take in the current message and all previous utterances as contexts and output a response which is appropriate given the entire conversation session. Recently, methods are proposed to capture long-span dependencies in contexts by concatenation [18,27], latent variable-based models

[8,13] or hierarchical approaches [12,25]. In this paper, we target at multi-turn conversations.

Most previous studies focus on two-party conversations, while the group conversation is a more general case of multi-turn conversations which involve more than two participants. It is more difficult to understand group conversations. Ouchi et al. [9] proposed to select the responses associated with addressees in multi-party conversations, which is basically a retrieval-based model by matching. Zhang et al. [32] extended the work by introducing an interactive neural network modeling, which is also for retrieval-based matching.

None of the work focuses on generation-based group conversations. More importantly, none of the work incorporates structure-aware formulation for group conversation models.

3 Group Conversation Model

In this section, we introduce our model for group conversations. First, we need to organize the conversation session according to the responding structures among users. We propose to construct the conversation context into a tree structure. With the established tree structure, we encode information for generation. Finally, we generate the target utterance in the decoding process.

Our problem formulation is straightforward. Given a group conversation with T utterances, we denote $X = \{X_i\}_{i=1}^{T}$. Each X_i is an utterance sentence. The goal is to generate the next utterance $Y = (y_1, y_2, \ldots, y_m)$ by estimating the probability $p(Y|X) = \prod p(y_t|y_{<t}, X)$.

Structure information is vital for group conversations. With different responding relationships, we construct different tree structures and accordingly, encode different information for the group conversations. Since we model the conversations as *trees*, we add the utterances with direct responding relationships onto the same branch of the tree. In this way, different responding relationships lead to different tree-branch structures.

Tree-based Formulation. Given the responding relationship among users, it is straightforward to establish the tree. If an utterance X_i is responding to the utterance X_j where $i > j$, we add an edge between X_i and X_j. X_j is the parent node of X_i while X_i is the child node of X_j. Generally, each utterance responds to one utterance but multiple subsequent utterances can respond to the same utterance. Suppose X_i and X_k both respond to X_j where $i > j$ and $k > j$. In this case, X_i and X_k are sibling nodes at the same level with a common parent node X_j.

As illustrated in Table 1, sometimes an utterance is addressed to a particular user *explicitly*. In this situation, we establish the tree without any ambiguity. In other cases, an utterance is not explicitly addressed to any user. To make the model practical, we introduce an assumption that if not explicitly designated, the utterance is addressing to the most recent utterance in the context. It is a simple assumption but holds in majority circumstances. For the group conversation in Table 1, we establish the tree in Fig. 1(b).

Splitting. Given a tree-structured conversation session, there are multiple sequences with shared nodes. We split the multiple sequences into separate sequences by duplicating the shared nodes, which is shown in Fig. 1(c). In this way, the conversation is represented by multiple sequences. Sequences have unique advantages over trees in batch learning. We identify which sequence the target utterance will be addressed to, and learn the embeddings of utterances along this sequence. We decode the target utterance based on the learned representation. Utterances from other sequences (i.e., other branches) will not be used for context information encoding and decoding.

Hierarchical Encoding. Our model is based on the encoder-decoder framework using the sequence-to-sequence model [19]. We implement with gated recurrent units (GRU) [3]. The encoder converts a sequence of embedding inputs $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ to hidden representations $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$ by:

$$\mathbf{h}_t = \text{GRU}(\mathbf{h}_{t-1}, \mathbf{x}_t) \tag{1}$$

Our model is established based on the hierarchical representations [6,12]. A hierarchical model draws on the intuition that just as the integration of words creates the overall meaning of an utterance, and furthermore the integration of multiple utterances creates the overall meaning of several utterances. To be more specific, we first obtain representation vectors at the utterance level by putting one layer of a recurrent neural network with GRU units on top of its containing words. The vector output at the ending time-step is used to represent the entire utterance sentence.

To build the representation for multiple utterances along a branch, another layer of GRU is placed on top of all utterances, computing representations sequentially for each time step. Representation computed at the final time step is used to represent the long sequence (i.e., the tree branch). Thus one GRU operates at the word-level, leading to the acquisition of utterance-level representations that are used as inputs into the second GRU that acquires the overall representations.

Decoding. After we obtain the encoded information, the decoder takes as input a context vector $\mathbf{c}_t$ and the embedding of a previously decoded word $\mathbf{y}_{t-1}$ to update its state st using another GRU:

$$\mathbf{s}_t = \text{GRU}(\mathbf{s}_{t-1}, [\mathbf{c}_t; \mathbf{y}_{t-1}]) \tag{2}$$

$[\mathbf{c}_t; \mathbf{y}_{t-1}]$ is the concatenation of the two vectors, serving as the input to GRU units. The context vector $\mathbf{c}_t$ is designed to dynamically attend on important information of the encoding sequence during the decoding process [1]. Once the state vector $\mathbf{s}_t$ is obtained, the decoder generates a token by sampling from the output probability distribution $\mathbf{o}_t$ computed from the decoder's state $\mathbf{s}_t$:

$$\begin{aligned} y_t \sim \mathbf{o}_t &= p(y_t | y_1, y_2, \ldots, y_{t-1}, \mathbf{c}_t) \\ &= \text{softmax}(\mathbf{W}_o \mathbf{s}_t) \end{aligned} \tag{3}$$

Table 2. Experimental results of different models based on automatic evaluations.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
NRM	9.85	3.04	1.38	0.67	3.98
C-Seq2Seq	10.45	4.13	2.08	1.02	3.43
HRED	11.23	4.40	2.45	1.42	4.38
Our method	**11.73**	**6.06**	**4.28**	**3.29**	**4.86**

4 Experiments

Data. We run experiments using the public Ubuntu Corpus[1] [8] for training and testing. The original data comes from the logs of Ubuntu IRC chat room where users discuss technical problems related to Ubuntu. The corpus consists of huge amount of records including over 7 million response utterances and 100 million words. We organize the dataset as tree-structured samples of 380k conversation sessions.

To be more specific, we take the last utterance in each session as the target to be generated and other utterances as inputs. We randomly divide the corpus into train-dev-test sets: 5,000 sessions for validation, 5,000 sessions for testing and the rest for training. We report results on the test set.

Baselines. We evaluate our model against a series of baselines. We include the context-insensitive baseline and context-aware methods (either non-hierarchical or hierarchical).

- *NRM.* Shang et al. [14] proposed the single-turn conversational model without contexts incorporated, namely neural responding machine (NRM). For NRM, only the penultimate utterance is used to generate the last utterance. It is performed using the Seq2Seq model with attention.
- *Context-Seq2Seq.* The context-sensitive seq2seq means that given a session, we use the last utterance as the target and all other utterances as the inputs. We concatenate all input utterances into a long utterance [18]. The concatenated contexts do not distinguish word or sentence hierarchies.
- *HRED.* The Hierarchical Recurrent Encoder-Decoder (HRED) model is a strong context-aware baseline which consists both word-level encoders and sentence-level encoders [12]. In this way, context utterances are encoded in two hierarchies as the training data.

None of these models takes the structure in group conversations into account. Our model incorporates structures into the hierarchical context-aware conversational model, where indicates a new insight.

Evaluation Metrics. We use the evaluation package released by [2] to evaluate our model and baselines. The package includes BLEU-1 to 4 [10] and

[1] http://dataset.cs.mcgill.ca/ubuntu-corpus-1.0/.

METEOR [4]. All these metrics evaluate the word overlap between the generated utterances and the ground truth targets. Still, note that these evaluation metrics have plenty room for improvement as to dialogue evaluations [7].

Results and Analysis. Table 2 shows the evaluation results. We observe that the performance is improved incrementally. From NRM to C-Seq2Seq, the improvement may be ascribed that context information is important for conversations with more than one turn. It concurs with many previous studies [25,27]. Hierarchical information depicted by fine-grained representation learning is also demonstrated to be useful [12,22]. None of the baselines formulates structure information, while our method utilizes a tree-based frame. Our method outperforms baselines in all metrics: structure-aware information is shown to be effective in group conversations.

Note that in our method, after splitting the tree into multiple sequences, we actually discard part of the context utterances during the encoding process. It is surprising that the model achieves even better results. We understand that within a group conversation, only the relevant information is useful to generate the target utterance. Irrelevant utterances on the other branches of the tree (i.e., other sequences) might be the noises for generation. It is interesting to see that "less becomes more" in group conversations.

5 Conclusion

In this paper, we proposed a tree-based model frame for structure-aware group conversations. According to different responding relations, we organize the group conversation as a tree with different branches involving multiple conversation threads. We split the established tree into multiple sequences, and we only use the target sequence to generate the next utterance. This method is quite simple but rather effective. We have performance improvement in terms of automatic evaluations, which indicate less context information results in better generations in group conversations. In other words, "less" is "more".

Acknowledgments. We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Key Research and Development Program of China (No. 2017YFC0804001), the National Science Foundation of China (NSFC No. 61672058). Rui Yan was sponsored by CCF-Tencent Open Research Fund and Microsoft Research Asia (MSRA) Collaborative Research Program.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
2. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: data collection and evaluation server. CoRR abs/1504.00325 (2015)

3. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: ICLR (2015)
4. Denkowski, M.J., Lavie, A.: Meteor universal: language specific translation evaluation for any target language. In: WMT@ACL (2014)
5. Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., Dolan, B.: A persona-based neural conversation model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 994–1003 (2016)
6. Li, J., Luong, T., Jurafsky, D.: A hierarchical neural autoencoder for paragraphs and documents. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1106–1115 (2015)
7. Liu, C.W., Lowe, R., et al.: How not to evaluate your dialogue system: an empirical study of unsupervised evaluation metrics for dialogue response generation. In: EMNLP 2016, pp. 2122–2132 (2016)
8. Lowe, R.J., Pow, N., Serban, I., Pineau, J.: The ubuntu dialogue corpus: a large dataset for research in unstructured multi-turn dialogue systems. In: SIGDIAL Conference (2015)
9. Ouchi, H., Tsuboi, Y.: Addressee and response selection for multi-party conversation. In: EMNLP, pp. 2133–2143 (2016)
10. Papineni, K., Roucos, S.E., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL (2002)
11. Qiu, M., Li, F.L., Wang, S., Gao, X., Chen, Y., Zhao, W., Chen, H., Huang, J., Chu, W.: AliMe chat: a sequence to sequence and rerank based chatbot engine. In: ACL 2017, pp. 498–503 (2017)
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: AAAI, pp. 3776–3784 (2016)
13. Serban, I.V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A.C., Bengio, Y.: A hierarchical latent variable encoder-decoder model for generating dialogues. In: AAAI 2017, pp. 3295–3301 (2017)
14. Shang, L., Lu, Z., Li, H.: 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 (Volume 1: Long Papers), vol. 1, pp. 1577–1586 (2015)
15. Song, Y., Li, C.T., Nie, J.Y., Zhang, M., Zhao, D., Yan, R.: An ensemble of retrieval-based and generation-based human-computer conversation systems. In: IJCAI 2018 (2018)
16. Song, Y., Tian, Z., Zhao, D., Zhang, M., Yan, R.: Diversifying neural conversation model with maximal marginal relevance. In: IJCNLP 2017, pp. 169–174 (2017)
17. Song, Y., Yan, R., Feng, Y., Zhang, Y., Zhao, D., Zhang, M.: Towards a neural conversation model with diversity net using determinantal point processes. In: AAAI 2018, pp. 5932–5939 (2018)
18. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversational responses. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 196–205 (2015)
19. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)

20. Tao, C., Gao, S., Shang, M., Wu, W., Zhao, D., Yan, R.: Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In: IJCAI 2018 (2018)
21. Tao, C., Mou, L., Zhao, D., Yan, R.: RUBER: an unsupervised method for automatic evaluation of open-domain dialog systems. In: AAAI 2018, pp. 722–729 (2018)
22. Tian, Z., Yan, R., Mou, L., Song, Y., Feng, Y., Zhao, D.: How to make context more useful? An empirical study on context-aware neural conversational models. In: Annual Meeting of the Association for Computational Linguistics, pp. 231–236 (2017)
23. Vinyals, O., Le, Q.: A neural conversational model. arXiv preprint arXiv:1506.05869 (2015)
24. Wang, H., Lu, Z., Li, H., Chen, E.: A dataset for research on short-text conversations. In: EMNLP, pp. 935–945 (2013)
25. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 496–505 (2017)
26. Yan, R.: "Chitty-Chitty-Chat Bot": Deep learning for conversational AI. In: IJCAI 2018 (2018)
27. Yan, R., Song, Y., Wu, H.: Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 55–64. ACM (2016)
28. Yan, R., Song, Y., Zhou, X., Wu, H.: Shall i be your chat companion? Towards an online human-computer conversation system. In: CIKM 2016, pp. 649–658 (2016)
29. Yan, R., Zhao, D.: Coupled context modeling for deep chit-chat: towards conversations between human and computer. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2018)
30. Yan, R., Zhao, D., E., W.: Joint learning of response ranking and next utterance suggestion in human-computer conversation system. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017, pp. 685–694 (2017)
31. Yao, L., Zhang, Y., Feng, Y., Zhao, D., Yan, R.: Towards implicit content-introducing for generative short-text conversation systems. In: EMNLP 2017, pp. 2190–2199 (2017)
32. Zhang, R., Lee, H., Polymenakos, L., Radev, D.: Addressee and response selection in multi-party conversations with speaker interaction RNNs. In: AAAI (2018)
33. Zhou, X., Dong, D., Wu, H., Zhao, S., Yu, D., Tian, H., Liu, X., Yan, R.: Multi-view response selection for human-computer conversation. In: EMNLP 2016, pp. 372–381 (2016)