TSDG: Content-aware Neural Response Generation with Two-stage Decoding Process

Junsheng Kong[*], **Zhicheng Zhong**[*], **Yi Cai**[†], **Xin Wu** and **Da Ren**
School of Software Engineering, South China University of Technology, Guangzhou, China
Key Laboratory of Big Data and Intelligent Robot (South China University of Technology), Ministry of Education
sescut_kongjunsheng@mail.scut.edu.cn, ycai@scut.edu.cn

Abstract

Neural response generative models have achieved remarkable progress in recent years but tend to yield irrelevant and uninformative responses. One of the reasons is that encoder-decoder based models always use a single decoder to generate a complete response at a stroke. This tends to generate high-frequency function words with less semantic information rather than low-frequency content words with more semantic information. To address this issue, we propose a content-aware model with two-stage decoding process named Two-stage Dialogue Generation (TSDG). We separate the decoding process of content words and function words so that content words can be generated independently without the interference of function words. Experimental results on two datasets indicate that our model significantly outperforms several competitive generative models in terms of automatic evaluation and human evaluation.

1 Introduction

With the development of deep learning, the open-domain neural response generation has achieved remarkable progress (Li et al., 2016; Serban et al., 2017b; Chen et al., 2019) in resent years. At present, most of generative models are based on encoder-decoder framework (Cho et al., 2014; Shang et al., 2015). In the decoding process, these models always use a single decoder to generate the final response at a stroke in a left-to-right manner. However, we find it hard for these methods to model the dependency of semantic between post and response which causes irrelevant and uninformative responses. We analyze this problem from the perspective of linguistics as following.

In linguistics, there are two different types of words to form a sentence, namely *content words* (words which have substantive lexical content) and *function words* (words which essentially serve to make grammatical properties) (Hill, 1952). For the response "I am going to read an interesting book." in Figure 1, content words "read, interesting, book" give us the most important semantic information which establishes the semantic dependency with the post, while function words "I, am, going, to, an" are used to stitch content words together. High-quality content words are a critical component of a relevant and informative response. Although function words are small in numbers (less than 0.04% of our vocabulary), they account for over half of the words used in our daily speech (Rochon et al., 2000). Therefore, function words are always high-frequency relative to the content words.

Figure 1: An example of content-aware response generation.

In vanilla encoder-decoder models, these models always use a single decoder to generate a complete response at a stroke. When the decoder generates content words and function words at a stroke, it tends to generate high-frequency function words with less semantic information rather than low-frequency content words with more semantic information. Since function words have very little substantive meaning, they not only are redundant for understanding semantic dependency, but also make the dependency sparse. Therefore, generating content words and function words at a stroke

[*] Both of authors contributed equally to this research.
[†] Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2121–2126

makes it difficult to learn the semantic dependency between the post and response.

To address the aforementioned issue, we propose a novel content-aware TSDG model with a two-stage decoding process. As shown in Figure 1, the key idea is to separate the decoding process of content words and function words so that content words can be generated independently without the interference of function words. In the first decoding stage, we use the first decoder to focus on generating a content word sequence according to the post. In the second decoding stage, we use the second decoder to expand the content word sequence to a complete and fluent response. Through this stage, our model gets the final fluent response including more relevant and informative content words.

Our contributions in this paper are two-fold:

(1) This paper analyzes the limitation of the encoder-decoder models which use a single decoder to generate a complete response at a stroke. For this limitation, we elaborate a content-aware TSDG model to generate a more informative and relevant response.

(2) Experimental results on two datasets demonstrate that our model can generate more appropriate content words and significantly outperforms several competitive generative models in terms of automatic evaluation and human evaluation.

2 Related Work

Open-domain conversation has long attracted the attention of researchers. The generative models have shown great potential in terms of flexibility, which has aroused a research hotspot. Most of generative models are based upon encoder-decoder framework (Cho et al., 2014; Shang et al., 2015). However, the traditional encoder-decoder models tend to generate short and uninformative responses, which are known as "safe responses" (Gao et al., 2019).

Lots of models have been proposed to solve this issue: (1) Modifying the objective function to penalize the generation probability of the safe response (Li et al., 2016). (2) Generating from latent variables to increase the diversity of response (Zhao et al., 2017; Serban et al., 2017b). (3) Using additional topic content (Xing et al., 2017). (4) Content-introducing methods (Mou et al., 2016; Yao et al., 2017). (5) Knowledge-based methods (Zhou et al., 2018; Tong et al., 2019). Zhou et al. (2018) take commonsense knowledge into account to facilitate conversation understanding. In the decoding process, these models always use a single decoder to generate the final response at a stroke in a left-to-right manner.

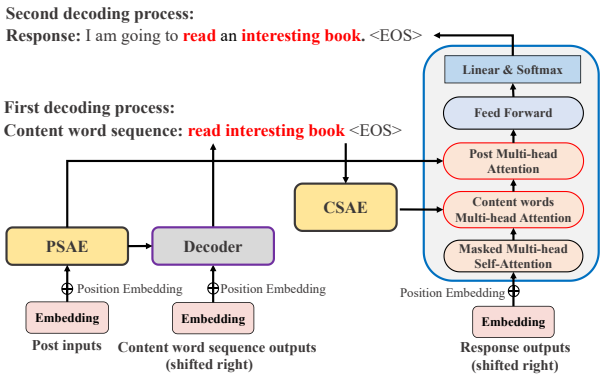

Figure 2: The architecture of our proposed model.

3 Model

The architecture of the proposed TSDG model is illustrated in Figue 2. It consists of an encoding process and a two-stage decoding process. Given a post $U = u_1, u_2..., u_I$ as input, our model first uses a Self-Attention Encoder (SAE) to encode them into a hidden vector. Then, the first decoding stage decodes this hidden vector into a content word sequence $C = c_1, c_2, ..., c_K$ without the influence of function word. Finally, the second decoding stage expands the content word sequence into a complete response $R = r_1, r_2, ..., r_J$.

3.1 Encoding process

In the encoding process, we use SAE to encode the utterance. SAE is a transformer encoder (Vaswani et al., 2017). There are two encoders in the encoding process: Post Self Attention Encoder (PSAE) and Content words Self Attention Encoder (CSAE) which encodes the post utterance and the content word sequence generated by first decoding process independently. The input (In_s) of the encoder is a sequence of word embedding with positional encoding added (Vaswani et al., 2017). We use $PSAE(U)$ to denote the process of encoding the post utterance and use $CSAE(C)$ to denote the process of encoding the content word sequence.

3.2 Two-stage decoding process

First decoding stage: Based on the hidden vector encoded by $PSAE(U)$, the first decoding stage uses a transformer decoder (Vaswani et al., 2017)

to generate the content words of response. When generating the i^{th} content word c_i, we have the generated words $c_{\leq i-1}$ as input. We use $In_c^{(i-1)}$ to denote the matrix representation of $c_{\leq i-1}$. The probabilities of the content word c_i decoded by the first decoding stage:

$$P(c_i) = Decoder(In_c^{(i-1)}, PSAE(U)) \quad (1)$$

The loss of the first decoding stage:

$$\mathcal{L}_1 = -\sum_{i=1}^{K}(logP(c_i)) \quad (2)$$

In the training process, we apply a rule-based content word extractor to automatically extract content words from the response in terms of Part-Of-Speech features and a stop word list. Based on the characteristic of the content words, its Part-Of-Speech should be noun, verb, adjective or adverb and not in the stop word list. Then, we take this content word sequence as ground truth to train the first decoding stage.

Second decoding stage: the second decoding stage aims to expand the content word sequence to a complete response. To capture the information of content word sequence and post, we propose a multi-layer multi-head attention decoder.

When generating the i^{th} word r_i of response, we have the generated words $r_{\leq i-1}$ as input. We use $In_r^{(i-1)}$ to denote the matrix representation of $r_{\leq i-1}$.

The first sub-layer is a multi-head self-attention:

$$G^{(i)} = M(In_r^{(i-1)}, In_r^{(i-1)}, In_r^{(i-1)}) \quad (3)$$

The second sub-layer is a content word multi-head attention:

$$H_{cw}^{(i)} = M(G^{(i)}, CSAE(C), CSAE(C)) \quad (4)$$

The third sub-layer is a post multi-head attention:

$$H_c^{(i)} = M(H_{cw}^{(i)}, PSAE(U), PSAE(U)) \quad (5)$$

The forth sub-layer is a position-wise fully connected feed-forward network:

$$F^{(i)} = FFN(H_c^{(i)}) \quad (6)$$

We use softmax to get the probabilities of the words decoded by the second decoding stage:

$$P(r_i) = softmax(F^{(i)}) \quad (7)$$

where r_i is the i^{th} word of response.

The loss of the second decoding stage :

$$\mathcal{L}_2 = -\sum_{i=1}^{J}(logP(r_i)) \quad (8)$$

Note that residual connection and layer normalization are used in each sub-layer, which are omitted in the presentation for simplicity.

During the training process, the total loss function of our model is a combination of $\mathcal{L}_1$ and $\mathcal{L}_2$:

$$\mathcal{L}_{total} = \mathcal{L}_1 + \lambda\mathcal{L}_2 \quad (9)$$

where λ ($\lambda > 0$) acts as a trade-off between the two items. We set λ to 1 in our experiment.

4 Experiments

4.1 Dataset

We conduct experiments on two datasets, namely the STC-SeFun dataset and the Weibo dataset.

STC-SeFun: A Short-Text Conversation dataset, in which each sentence segment in the query-response pairs is labeled with its sentence functions (Bi et al., 2019). There are 45,022 post-response pairs for training, 9,590 for validation, and another unseen 9,590 samples for testing.

Weibo: A high-quality Weibo conversation pairs pre-processed by (Gao et al., 2019) from the benchmark dataset (Shang et al., 2015). We used 50,000 post-response pairs to train the model. We use another unseen 997 and 800 samples for validation and testing, respectively.

As pre-processing, we remove duplicate pairs and the pairs with a post or a response having less than 2 words. We also truncate the sentences with more than fifty characters.

4.2 Baselines

- Seq2Seq-atte: a basic Seq2Seq neural response generative model (Shang et al., 2015) with global attention. We use a Seq2Seq model implemented by OpenNMT[1].
- MrRNN: a content-introducing model based on Seq2Seq (Serban et al., 2017a). We re-implemented this work to get the results.
- MMPMS: the model with the state-of-the-art performance on the Short text Conversation (STC) task (Chen et al., 2019). We re-run

[1] https://github.com/OpenNMT/OpenNMT-py

Dataset	Models	Automatic Evaluation			Human Evaluation		
		BLEU1	BLEU2	CWS	Fluency	Informativeness	Relevance
STC-SeFun	Seq2Seq-atte	0.1404	0.1057	2.3221	1.440	0.950	0.830
	Seq2Seq-trans	0.1502	0.1135	2.2860	1.490	0.993	1.116
	MrRNN	0.1596	0.1206	2.3755	1.517	0.99	1.013
	MMPMS	0.1282	0.0985	2.5070	1.233	0.977	0.423
	Skeleton	0.1572	0.1189	2.2616	1.560	0.963	0.910
	TSDG	**0.1705**	**0.1312**	**2.8822**	**1.677**	**1.277**	**1.133**
	Ground truth	-	-	3.2033	1.863	1.436	1.680
Weibo	Seq2Seq-atte	0.1377	0.1107	3.1225	0.887	0.627	0.263
	Seq2Seq-trans	0.0941	0.0750	2.8550	1.370	0.810	0.447
	MrRNN	0.1503	0.1214	3.7200	1.260	0.770	0.423
	MMPMS	0.1360	0.1102	3.7175	1.020	0.603	0.213
	Skeleton	0.1454	0.1169	3.5238	1.227	0.497	0.213
	TSDG	**0.1657**	**0.1360**	**4.1562**	**1.636**	**0.933**	**0.517**
	Ground truth	-	-	6.9038	1.877	1.877	1.730

Table 1: The experimental results of automatic and human evaluation.

the released code[2] to obtain the results on our dataset.

- Skeleton: a model (Cai et al., 2019) to enhance generative models with information retrieval technologies for dialogue response generation. We re-run the released code[3] to obtain the results on our dataset.

- Seq2Seq-trans: an ablated model of TSDG. We replace the two-stage decoding process in TSDG with a basic transformer decoder to directly generate the response.

4.3 Implementation Details

In our experiments, we use OpenNMT-py (Klein et al., 2017) as the code framework of TSDG. The layers of both encoder and decoder are set to 3. The number of attention heads in multi-head attention is 8 and the filter size is 2048. The dimension of word embedding is set to 512 empirically. We use Adam for optimization. When decoding both in two stages, the beam size is set to 5. The experiments are conducted on an NVIDIA 2080 Ti.

4.4 Automatic and Human Evaluation

Automatic Evaluation: We adopt BLEU1 and BLEU2 to automatically evaluate the response generation performance by nltk package [4]. To evaluate the quantity of content words, we use the content words score (CWS):

$$CWS = \frac{\sum_{i=1}^{T} n_i}{T} \tag{10}$$

where T denotes the size of the test set, n_i denotes the number of content words in i^{th} predicted response.

Human Evaluation: Human evaluations are essential for response generation. We randomly sampled 100 utterances from the test set. We asked 3 experienced annotators to score the fluency, relevance and informativeness of responses.

4.5 Results and Analysis

Table 1 shows the results of automatic and human evaluation. TSDG outperforms all baseline methods both on automatic and human evaluation, and the improvement is significant in a statistical sense (p-value < 0.01). This indicates that TSDG generates a more appropriate response in terms of fluency, informativeness and relevance. The performance of the ablated model (Seq2seq-trans) suffers from the ablation, which demonstrates that the two-stage decoding process is essential for TSDG.

We find that the CWSs of baselines are significantly lower than the CWS of ground truth. The significant improvement in CWS indicates that TSDG can effectively increase the proportion of content words in the response. We also compare the content words generated by our model with other models side-by-side on 100 test cases which are randomly picked from STC-SeFun dataset. The human evaluation results are shown in Table 2. Note that our model consistently outperforms the comparison models with a large margin. This superior performance confirms that our model can generate more appropriate content words.

[2] https://github.com/PaddlePaddle/models/tree/develop/PaddleNLP/Research/IJCAI2019-MMPMS.

[3] https://github.com/jcyk/Skeleton-to-Response

[4] http://www.nltk.org/_modules/nltk/translate/bleu_score.html

	Ours Better(%)	Tie(%)	Ours Worse(%)
Seq2Seq-atte	48.0	36.0	16.0
Seq2Seq-trans	24.0	65.3	10.7
MrRNN	32.0	49.3	18.7
MMPMS	40.0	48.0	12.0
Skeleton	48.0	41.3	10.7

Table 2: Experimental results about content words.

To further evaluate the relevance between two decoding stages, we use the content words acc (CWA):

$$CWA = \frac{\sum_{i=1}^{T} \frac{k_i}{l_i}}{T} \quad (11)$$

where T denotes the size of test set, k_i is the number of content words which both in the i^{th} content word sequence and i^{th} predicted response, l_i is the number of i^{th} content word sequence predicted by the first stage. The higher the CWA, the higher the relevance between the two stages. Under two datasets, our model gets CWA of 0.9252 and 0.9152 separately. Both of them are higher than 0.9, which verifies that our model can make good use of first decoded content. There still is some room for improvement.

To show the influence of the content word sequence more clearly, we feed different content word sequences into the second decoding stage to compare the generated response. The results are shown in Table 3. These examples demonstrate that content words generated by the first decoding stage play an important role in the generation of final response.

Post: 我喜欢深圳 (I like Shenzhen)	
Given content words	TSDG response
喜欢 like	我也**喜欢** I also **like**
喜欢 深圳 like Shenzhen	我也**喜欢深圳** I also **like Shenzhen**
喜欢 深圳 感觉 包容 城市 like Shenzhen think inclusive city	**喜欢**+1 **感觉**是个很**包容**的**城市** **Like** +1 I **think** it is a **inclusive city**.

Table 3: Examples from STC-SeFun dataset.

5 Conclusion

In this paper, we analyze the limitation of the current generative models in the decoding process. To address this, we propose a content-aware neural response generative model with a two-stage decoding process. Evaluation results on two datasets indicate that our model can generate more appropriate content words and significantly outperform several competitive models in terms of automatic and human evaluation. There still is some room for improvement. We will refine our model from two decoding stages independently in the future.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities, SCUT (No. 2017ZD048, D2182480), the Science and Technology Planning Project of Guangdong Province (No.2017B050506004), the Science and Technology Programs of Guangzhou (No.201704030076, 201802010027,201902010046) and the National Natural Science Foundation of China (62076100).

References

Wei Bi, Jun Gao, Xiaojiang Liu, and Shuming Shi. 2019. Fine-grained sentence functions for short-text conversation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3984–3993. Association for Computational Linguistics.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2019. Skeleton-to-response: Dialogue generation guided by retrieval memory. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1219–1228. Association for Computational Linguistics.

Chaotao Chen, Jinhua Peng, Fan Wang, Jun Xu, and Hua Wu. 2019. Generating multiple diverse responses with multi-mapping and posterior mapping selection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4918–4924. ijcai.org.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, and Shuming Shi. 2019. Generating multiple diverse responses for short-text conversation. In *The Thirty-Third AAAI Conference on Artificial Intelligence,*

AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 6383–6390. AAAI Press.

Archibald A Hill. 1952. The structure of english, an introduction to the construction of english sentences.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3349–3358. ACL.

Elizabeth Rochon, Eleanor M Saffran, Rita Sloan Berndt, and Myrna F Schwartz. 2000. Quantitative analysis of aphasic sentence production: Further development and new data. *Brain and language*, 72(3):193–218.

Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. 2017a. Multiresolution recurrent neural networks: An application to dialogue response generation. In (Singh and Markovitch, 2017), pages 3288–3294.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017b. A hierarchical latent variable encoder-decoder model for generating dialogues. In (Singh and Markovitch, 2017), pages 3295–3301.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. 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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1577–1586. The Association for Computer Linguistics.

Satinder P. Singh and Shaul Markovitch, editors. 2017. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. AAAI Press.

Peihao Tong, Qifan Zhang, and Junjie Yao. 2019. Leveraging domain context for question answering over knowledge graph. *Data Sci. Eng.*, 4(4):323–335.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In (Singh and Markovitch, 2017), pages 3351–3357.

Lili Yao, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2190–2199. Association for Computational Linguistics.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654–664. Association for Computational Linguistics.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4623–4629. ijcai.org.