Modeling Compositionality with Dependency Graph for Dialogue Generation

Xiaofeng Chen[1], Yirong Chen[1], Xiaofen Xing[1]*, Xiangmin Xu[1], Wenjing Han[1], Qianfeng Tie[1]
[1]UBTECH-SCUT Joint Research Lab, School of Electronic and Information Engineering,
South China University of Technology, China
{eexiaofengchen, eeyirongchen, eewenjinghh, 202120112795}@mail.scut.edu.cn
{xmxu, xfxing}@scut.edu.cn

Abstract

Because of the compositionality of natural language, syntactic structure which contains the information about the relationship between words is a key factor for semantic understanding. However, the widely adopted Transformer is hard to learn the syntactic structure effectively in dialogue generation tasks. To explicitly model the compositionaity of language in Transformer Block, we restrict the information flow between words by constructing directed dependency graph and propose Dependency Relation Attention (DRA). Experimental results demonstrate that DRA can further improve the performance of state-of-the-art models for dialogue generation.

1 Introduction

In natural language, complex semantics are often expressed by combining words with certain rules. For example, "room" can express higher-level semantics by fusing the information of "a" and "hotel", and the meaning of "reserve" will be clearer after fusing the information of "room". Prior works have achieved great success in NLP tasks by leveraging syntactic structure knowledge, such as semantic relatedness (Tai et al., 2015; Gupta and Zhang, 2018), sentiment analysis (Ma et al., 2015; Sun et al., 2019), relation extraction (Tian et al., 2021), and named entity recognition (Aguilar and Solorio, 2019; Xu et al., 2021).

Due to the strong ability to capture long-term dependencies (Tang et al., 2018), many recent works have adopted the Transformer block (Vaswani et al., 2017) to extract context features in dialogue generation tasks (Su et al., 2019; Liu et al., 2020; Song et al., 2021). However, it is hard for Transformer block to implicitly learn the compositionality of language in the training process of dialog generation, since it simply uses position embeddings to represent the relationships between words, and it learns the local position information that can only be effective in masked language modeling (Wang and Chen, 2020). Besides, the computation of attention weights on unrelated word pairs in Transformer block is redundant and decreases performance.

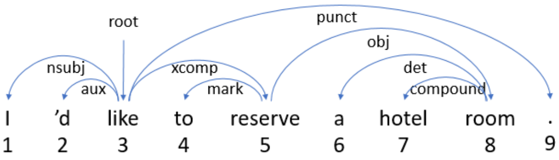


Figure 1: An example of dependency graph.

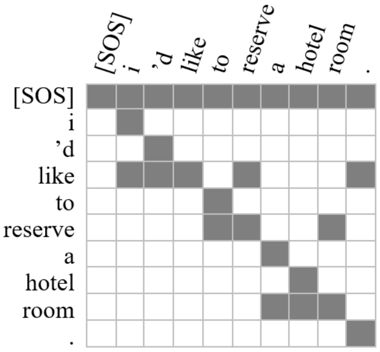
To obtain better distributed representations of context in dialogue generation tasks, we propose Dependency Relation Attention to model the relationship between words as an alternative to position embeddings. Specifically, we incorporate dependency relation knowledge that contains syntactic structure information into the Transformer block. As shown in Figure 1, we use the dependency parser (Chen and Manning, 2014) in the StanfordCoreNLP toolkit (Manning et al., 2014) to build dependency graphs of utterances. Then, the Dependency Relation Mask is generated to avoid performing attention on words without dependency relations, and the fusion of information among words depends on the direction specified by the dependency graph. Our contributions can be summarized as follows:

- We propose Dependency Relation Attention, a novel method for expressing relationships between words as an alternative to position embeddings.

- We demonstrate that our method can further improve the performance of Transformer and DialogBERT (Gu et al., 2021) in dialogue generation task by conducting experiments on two datasets.

* Corresponding author: xfxing@scut.edu.cn

Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI), pages 9 - 16
July 14, 2022 ©2022 Association for Computational Linguistics



	[SOS]	i	'd	like	to	reserve	a	hotel	room	.
[SOS]	■	■	■	■	■	■	■	■	■	■
i		■								
'd			■							
like		■	■	■		■				■
to					■					
reserve					■	■			■	
a							■			
hotel								■		
room							■	■	■	
.										■

Figure 2: Dependency Relation Mask.

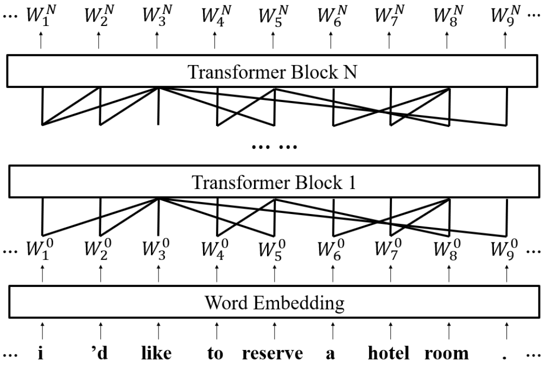
2 Related Works

In the past few years, dependency graph has drawn attention from many researchers in the field of NLP. Strubell et al. (2018) propose Syntactically-informed self-attention and incorporate syntactic dependency knowledge into a attention head of specific Transformer block. To make the attention learned by Transformer more interpretable, Wang et al. (2019) propose Constituent Attention which makes each position not attend to the position in different constituents. Ahmad et al. (2021) explicitly fuse structural information to learn the dependency relations between words with different syntactic distances.

In dialogue generation tasks, to improve the quality of generated responses, previous works focus on capturing the high-level relationships between contexts and responses (Xing et al., 2018; Zhang et al., 2019) or between utterances in context(Gu et al., 2021). How to effectively model the relationships between words in Transformer has not been explored. Inspired by TreeLSTM (Tai et al., 2015), our method aim at modeling the compositionality in language, then the Transformer block does not need to learn the relationships between words through position embeddings in the training process of dialog generation. The differences between DRA and others dependency relation-aware attention mechanisms are: (1) DRA incorporates the dependency arc directions into Transformer block to model the relationships between words instead of position embeddings. (2) The position embeddings are excluded for the models with DRA applied.

3 Method

In dialogue generation tasks, given a piece of context containing m utterances $U = \{X_1, ..., X_m\}$ as inputs, where $X_i = \{x_{i,1}, ..., x_{i,n_i}\}, i \in [1, m]$ indicates the i-th utterance containing n_i words, dialogue generation models map it into feature vectors and estimate the generation probability of the corresponding response $Y = \{y_1, ..., y_t\}$:



... W_1^N W_2^N W_3^N W_4^N W_5^N W_6^N W_7^N W_8^N W_9^N ...
Transformer Block N
... ...
Transformer Block 1
... W_1^0 W_2^0 W_3^0 W_4^0 W_5^0 W_6^0 W_7^0 W_8^0 W_9^0 ...
Word Embedding
... i 'd like to reserve a hotel room . ...

Figure 3: Illustration of applying DRA to standard Transformer encoder. Dependency Relation Mask is used to model the semantic relationship between words instead of position embeddings.

$$p(y_1, ..., y_t|U) = \prod_{k=1}^{t} p(y_k|y_{<k}, U) \quad (1)$$

To obtain a better representations of context, we incorporate dependency relation knowledge into the Transformer block, which is widely used in recent works.

3.1 Dependency Relation Mask

We use the StanfordCoreNLP toolkit[1] to parse the dependency relations and obtain a set of triples $R_{i,j} = (r_{i,j}, g_{i,j}, d_{i,j}), j \in [1, n_i]$ for each utterance, where $r_{i,j}$, $g_{i,j}$, and $d_{i,j}$ represent the name of the relation, the index of the governor, and the index of the dependent (the j-th word in the i-th utterance) respectively. For the utterance in Figure 1, here is the triples R returned from the parser:

$\bullet(nsubj, 3, 1)$ $\bullet(aux, 3, 2)$ $\bullet(ROOT, 0, 3)$
$\bullet(mark, 5, 4)$ $\bullet(xcomp, 3, 5)$ $\bullet(det, 8, 6)$
$\bullet(compound, 8, 7)$ $\bullet(obj, 5, 8)$ $\bullet(punct, 3, 9)$

The indexes in dependency relation triples $E = \{(g_1, d_1), ..., (g_n, d_n)\}$ are used to generate the Dependency Relation Mask $M \in \mathbb{R}^{(n+1)\times(n+1)}$. Figure 2 shows an example:

$$M_{u,v} = \begin{cases} 0, & u = 0 \quad or \quad u = v \\ 0, & (u, v) \in E \\ -\infty, & otherwise \end{cases} \quad (2)$$

3.2 Dependency Relation Attention

The main idea of our proposed method is to use Dependency Relation Attention (DRA) to model

[1] https://nlp.stanford.edu/software/nndep.html

the compositionality, instead of letting models implicitly learn the relationships between words through position embeddings. Figure 3 is an illustration of applying Dependency Relation Attention to a standard Transformer encoder. Specifically, for the l-th layer of the Transformer block in the encoding process, the hidden states of words $W^l \in \mathbb{R}^{n \times d_{hidden}}$ are linearly mapped to three subspaces in different heads of multi-head attention network: $Q^l \in \mathbb{R}^{n \times d_{head}}$, $K^l \in \mathbb{R}^{n \times d_{head}}$ and $V^l \in \mathbb{R}^{n \times d_{head}}$. The attention score matrix $S^l \in \mathbb{R}^{n \times n}$, which indicates the strength of relationships between words, is calculated by:

$$S^l = \frac{Q^l {K^l}^T}{\sqrt{d_{head}}} \tag{3}$$

Then, the attention scores of unrelated word pairs are masked:

$$S^l_{masked} = S^l + M \tag{4}$$

The hidden states of words W are updated based on the dependency relations:

$$\begin{aligned} A^l_{masked} &= softmax(S^l_{masked}) \\ O^{l,i} &= A^{l,i}_{masked} V^{l,i} \\ O^l &= concat(O^{l,1}, ..., O^{l,n_{head}}) \\ W^{l+1} &= W^l + O^l \end{aligned} \tag{5}$$

4 Experiments

4.1 Settings

4.1.1 Datasets

In our experiment, we use DailyDialog (Li et al., 2017) and EmpatheticDialogues (Rashkin et al., 2019) to verify the effectiveness of our method. They contains 11.1K, 1K, 1K and 19.5K, 2.7K, 2.5K dialogues for training, validation, testing, respectively. To accommodate the granularity of the word segmentation of the dependency parser and ensure fairness, StanfordCoreNLP toolkit is used to tokenize utterances for all models. Besides, we report the results of methods with subword tokenization in appendix. Words with word frequency less than 3 are replaced by "[UNK]". For each sample, dialogue turn and utterance length are limited to 4 and 50, respectively.

4.1.2 Compared Methods

We apply DRA to Transformer (Vaswani et al., 2017) and DialogBERT (Gu et al., 2021) and position embeddings are excluded. The performance of models before and after the modification and the following methods are compared: ReCoSa (Zhang et al., 2019), LISA (Strubell et al., 2018), Tree-Transformer (Wang et al., 2019) and GATE (Ahmad et al., 2021). Position embeddings are included for all baseline models.

We set the hidden sizes of all models to 768. The number of Transformer layers is set to 3. Each Transformer block contains 16 attention heads. The word embedding layers of all models are initialized with GloVe 300-dimensional word embeddings (Pennington et al., 2014). The batch size is set to 40. All models are trained by the AdamW (Loshchilov and Hutter, 2018) optimizer with weight decay of 0.01. We linearly warm up the learning rate from 0 to 5e-4 at the first 3000 steps. Afterward, the learning rate decreases to 0 linearly during training.

4.1.3 Evaluation Metrics

Automatic evaluation. PPL, BLEU (Papineni et al., 2002) and Distinct (Li et al., 2016) are employed to reflect the degree of fluency, relevance and diversity of generated responses respectively. They are widely used in dialog generation tasks (Song et al., 2020; Liang et al., 2021).

Human evaluation. We randomly select 100 contexts from the DailyDialog test set and generate responses with models trained on DailyDialog. Based on grammatical correctness and contextual coherence, three annotators are asked to score the generated responses independently with the following grading scale: "+0" (response is not fluent), "+1" (response is fluent but irrelevant), and "+2" (response is fluent and relevant).

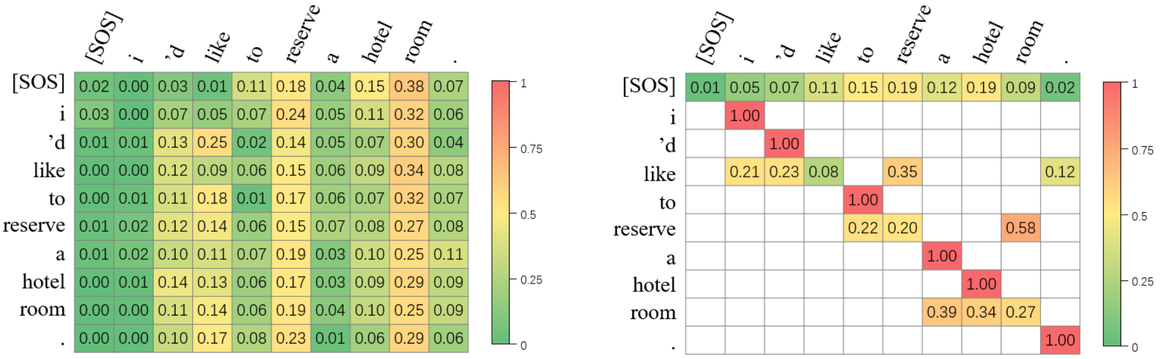
4.2 Experimental Results

Table 1 gives the automatic evaluation results on DailyDialog and EmpatheticDialogues validation set. For both datasets, Transformer+DRA and DialogBERT+DRA achieved the best performance on PPL and Dist-2 respectively. Transformer+DRA achieved comparable BLEU-2 scores in contrast to DialogBERT+DRA. It is worth noting that DRA improved the performance of Transformer and DialogBERT on all automatic metrics, which indicates that our method can help these two models generate more fluent, relevant, and diverse responses. We also study the computational efficiency and the impact of parsing errors, the results are shown in appendix.

The results of human evaluation are shown in

Model	DailyDialog			EmpatheticDialogues		
	PPL	BLEU-2	Dist-2	PPL	BLEU-2	Dist-2
ReCoSa	19.846	20.538	16.611	34.450	19.062	7.619
LISA	18.378	19.002	17.011	32.467	19.169	6.974
TreeTransformer	18.155	20.035	17.847	31.862	19.755	7.870
GATE ($\delta = 1$)	18.405	19.142	17.742	32.273	18.640	7.452
Transformer	18.278	19.519	17.381	32.329	18.553	7.499
Transformer+DRA	**17.628**	21.140	18.396	**31.604**	**19.966**	8.203
DialogBERT	20.056	18.069	15.562	35.643	17.199	5.064
DialogBERT+DRA	17.878	**21.786**	**21.283**	32.785	19.739	**9.601**

Table 1: Automatic evaluation results on DailyDialog and EmpatheticDialogues validation set.



	[SOS]	i	'd	like	to	reserve	a	hotel	room	.
[SOS]	0.02	0.00	0.03	0.01	0.11	0.18	0.04	0.15	0.38	0.07
i	0.03	0.00	0.07	0.05	0.07	0.24	0.05	0.11	0.32	0.06
'd	0.01	0.01	0.13	0.25	0.02	0.14	0.05	0.07	0.30	0.04
like	0.00	0.00	0.12	0.09	0.06	0.15	0.06	0.09	0.34	0.08
to	0.00	0.01	0.11	0.18	0.01	0.17	0.06	0.07	0.32	0.07
reserve	0.01	0.02	0.12	0.14	0.06	0.15	0.07	0.08	0.27	0.08
a	0.01	0.02	0.10	0.11	0.07	0.19	0.03	0.10	0.25	0.11
hotel	0.00	0.01	0.14	0.13	0.06	0.17	0.03	0.09	0.29	0.09
room	0.00	0.00	0.11	0.14	0.06	0.19	0.04	0.10	0.25	0.09
.	0.00	0.00	0.10	0.17	0.08	0.23	0.01	0.06	0.29	0.06

(a) Standard Transformer.

	[SOS]	i	'd	like	to	reserve	a	hotel	room	.
[SOS]	0.01	0.05	0.07	0.11	0.15	0.19	0.12	0.19	0.09	0.02
i		1.00								
'd			1.00							
like		0.21	0.23	0.08		0.35				0.12
to					1.00					
reserve					0.22	0.20			0.58	
a							1.00			
hotel								1.00		
room							0.39	0.34	0.27	
.										1.00

(b) Transformer+DRA.

Figure 4: The average attention weights of the last layer of Transformer encoder in different methods.

Model	+2	+1	+0	Avg.
ReCoSa	29.7	52.7	17.7	1.12
LISA	35.3	51.7	13.0	1.22
TreeTransformer	32.3	55.3	12.3	1.20
GATE ($\delta = 1$)	32.7	55.0	12.3	1.20
Transformer	33.3	54.3	12.3	1.21
Transformer+DRA	47.0	39.0	14.0	1.33
DialogBERT	32.3	59.3	8.3	1.24
DialogBERT+DRA	50.0	43.3	6.7	**1.43**

Table 2: Human evaluation results. (in %)

Table 2. The Fleiss' kappa score (Fleiss, 1971) for assessing agreement among annotators was 0.563, which can be interpreted as "moderate agreement". This shows that DRA can enhance the semantic understanding of Transformer block and help models generate more relevant responses, especially for the hierarchical Transformer encoder architecture.

4.3 Discussions

To further explore why our method can improve the performance of the Transformer encoder, we visualized the attention weights of the last layer of the Transformer encoder in different methods. Taking the utterance in Figure 1 as input, Figure 4 shows the mean value of attention weights of 16 heads in standard Transformer and Transformer+DRA. We can see that, in standard Transformer, the Transformer block assigns very similar weights to each part of the utterance when updating the hidden state of different words. This means that standard Transformer encoder can find the key parts of the utterance, but does not learn the relationships between words. In Transformer+DRA, for each word, attention weights are assigned to appropriate parts. For example, when updating the hidden state of "reserve", the Transformer block pays more attention to the "room" that has merged the information of "a" and "hotel". In other words, DRA makes it easier for Transformer encoder to understand the relationships between words and generate more meaningful distributed representations.

5 Conclusion and Future Work

In this paper, we propose Dependency Relation Attention (DRA) to model the relationships between words instead of position embeddings in the Transformer encoder. Experimental results show that our method can further improve the performance of models that use Transformer block to obtain the distributed representations of context in dialogue generation task. In the future, we will study the effect of the specific domains that parsers are usually trained in, as well as the possibility of improving the performance of pretrained language models with DRA.

6 Acknowledgement

The work is supported in part by the National Natural Science Foundation of China under Grant U1801262; in part by the Key-Area Research and Development Program of Guangdong under Grant 2019B010154003; in part by the Science and Technology Project of Guangzhou under Grant 202103010002; and in part by the Guangdong Provincial Key Laboratory of Human Digital Twin under Grant 2022B1212010004.

References

Gustavo Aguilar and Thamar Solorio. 2019. Dependency-aware named entity recognition with relative and global attentions. *arXiv e-prints*, pages arXiv–1909.

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. Gate: graph attention transformer encoder for cross-lingual relation and event extraction. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, volume 4, pages 74–75.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2021. Dialogbert: Discourse-aware response generation via learning to recover and rank utterances. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12911–12919.

Amulya Gupta and Zhu Zhang. 2018. To attend or not to attend: A case study on syntactic structures for semantic relatedness. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2116–2125.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Yunlong Liang, Fandong Meng, Ying Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. Infusing multi-source knowledge with heterogeneous graph neural network for emotional conversation generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13343–13352.

Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. 2020. You impress me: Dialogue generation via mutual persona perception. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1417–1427.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Mingbo Ma, Liang Huang, Bowen Zhou, and Bing Xiang. 2015. Dependency-based convolutional neural networks for sentence embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 174–179.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Haoyu Song, Yan Wang, Kaiyan Zhang, Wei-Nan Zhang, and Ting Liu. 2021. BoB: BERT over BERT for training persona-based dialogue models from limited personalized data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–177, Online. Association for Computational Linguistics.

Haoyu Song, Yan Wang, Weinan Zhang, Xiaojiang Liu, and Ting Liu. 2020. Generate, delete and rewrite: A three-stage framework for improving persona consistency of dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5821–5831.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance rewriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5679–5688, Hong Kong, China. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. 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)*, pages 1556–1566.

Gongbo Tang, Mathias Müller, Annette Rios Gonzales, and Rico Sennrich. 2018. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021. Dependency-driven relation extraction with attentive graph convolutional networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4458–4471, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019. Tree transformer: Integrating tree structures into self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1061–1070.

Yu-An Wang and Yun-Nung Chen. 2020. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6840–6849.

Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical recurrent attention network for response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lu Xu, Zhanming Jie, Wei Lu, and Lidong Bing. 2021. Better feature integration for named entity recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3457–3469.

Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019. ReCoSa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3721–3730, Florence, Italy. Association for Computational Linguistics.

A Additional Analysis

Extra experiments were conducted to further analyse models that applied with DRA, position embedding (*PE.*), or subword-level tokenization (*ST.*), since DRA and PE. can be applied to transformer block at the same time. The result is shown in Table 3 (*Trans.*, *Dial.*, *B-2* and *D-2* denote Transformer, DialogBERT, BLEU-2 and Dist-2, respectively). *ST.* can not improve the fluency (PPL) and diversity (D-2) of generated responses although it can promote higher BLEU score. Besides, the models with DRA can not handle the information of position embeddings well, we need to design some methods that can model the information of word order in dependency graph in the future.

Model	DailyDialog		
	PPL	B-2	D-2
Transformer	18.28	19.52	17.38
- Trans.+ST.	18.59	**22.15**	16.64
- Trans.+DRA	**17.63**	21.14	**18.40**
- Trans.+DRA+PE.	18.13	19.66	18.08
DialogBERT	20.06	18.07	15.56
- Dial.+ST.	20.07	19.51	15.72
- Dial.+DRA	**17.88**	**21.79**	**21.28**
- Dial.+DRA+PE.	20.32	17.86	15.77

Table 3: Result of extra comparison.

B Comparison of Running Time

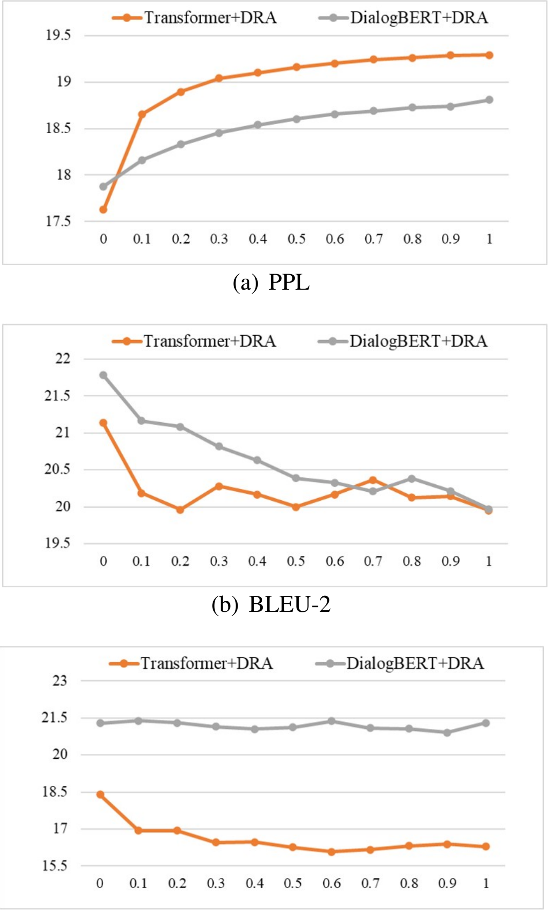
Table 4 shows the average time occupied by different models to generating response for each dialogue in DailyDialog (*Pre.* denote the process of word tokenization and dependency relation parsing of the raw text, *Gen.* denote the process of inference). We can see that the dependency parsing process does not take much time.

Model	Pre.	Gen.	Total
Transformer	0.005s	0.111s	0.116s
Transformer+DRA	0.028s	0.115s	0.143s
DialogBERT	0.005s	0.123s	0.128s
DialogBERT+DRA	0.030s	0.118s	0.148s

Table 4: Comparison of running time.

C Results of Parsing Errors.

As the accuracy of dependency parsing will affect the downstream task performance, it is worthwhile to investigate the result of the errors that result from syntactic parsing. We simulate parsing errors by manually changing the parsing results, specifically, the attention weights with dependency relations will be masked and those without dependency relations will not. Figure 5 show how the parsing errors affect PPL, BLEU-2, Dist-2 of models on DailyDialog validation set. The horizontal axis in the figure represents the proportion of parsing errors. It shows that our proposed method has certain robustness, especially for the hierarchical Transformer encoder architecture.



(a) PPL

(b) BLEU-2

(c) Dist-2

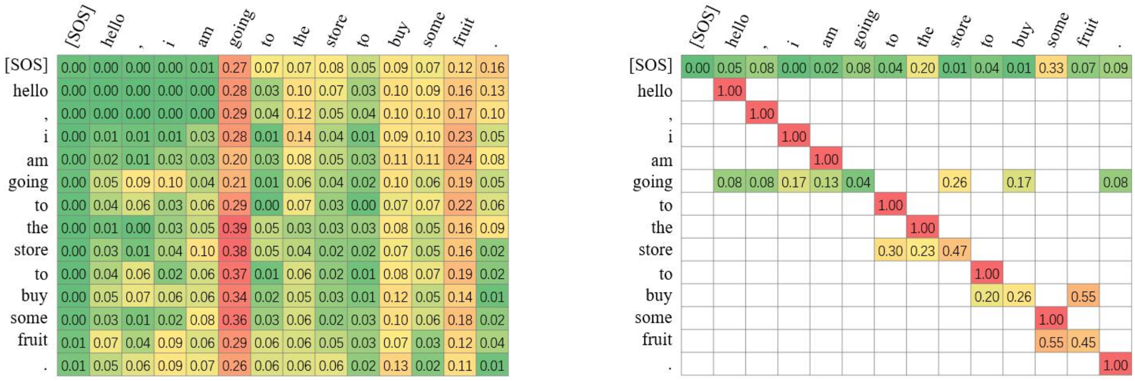
Figure 5: The result of parsing errors.

D Samples of Generated Dialogues

Table 5 and 6 provide some examples of the generated responses. The visual attention weights of different methods are presented in Figure 6 and 7. The models with DRA will focus on the relevant words when updating the hidden state of each word. They demonstrates that Dependency Relation Attention can help Transformer and DialogBERT generate better responses.

Example 1	
Speaker1:	Hello, Miao Li, where are you going?
Speaker2:	Hello, I am going to the store to buy some fruit.
Gold Resp:	Oh, would you do me a favor?
Transformer:	Oh, I'm afraid I'm going to take the train station.
Transformer+DRA:	What kind of fruit do you like?
DialogBERT:	Would you like some dessert?
DialogBERT+DRA:	What are you going to buy?

Table 5: Example responses from different models.



	[SOS]	hello	,	i	am	going	to	the	store	to	buy	some	fruit	.
[SOS]	0.00	0.00	0.00	0.00	0.01	0.27	0.07	0.07	0.08	0.05	0.09	0.07	0.12	0.16
hello	0.00	0.00	0.00	0.00	0.00	0.28	0.03	0.10	0.07	0.03	0.10	0.09	0.16	0.13
,	0.00	0.00	0.00	0.00	0.00	0.29	0.04	0.12	0.05	0.04	0.10	0.10	0.17	0.10
i	0.00	0.01	0.01	0.01	0.03	0.28	0.01	0.14	0.04	0.01	0.09	0.10	0.23	0.05
am	0.00	0.02	0.01	0.03	0.03	0.20	0.03	0.08	0.05	0.03	0.11	0.11	0.24	0.08
going	0.00	0.05	0.09	0.10	0.04	0.21	0.01	0.06	0.04	0.02	0.10	0.06	0.19	0.05
to	0.00	0.04	0.06	0.03	0.06	0.29	0.00	0.07	0.03	0.00	0.07	0.07	0.22	0.06
the	0.00	0.01	0.00	0.03	0.05	0.39	0.05	0.03	0.03	0.03	0.08	0.05	0.16	0.09
store	0.00	0.03	0.01	0.04	0.10	0.38	0.05	0.04	0.02	0.02	0.07	0.05	0.16	0.02
to	0.00	0.04	0.06	0.02	0.06	0.37	0.01	0.06	0.02	0.01	0.08	0.07	0.19	0.02
buy	0.00	0.05	0.07	0.06	0.06	0.34	0.02	0.05	0.03	0.01	0.12	0.05	0.14	0.01
some	0.00	0.03	0.01	0.02	0.08	0.36	0.03	0.06	0.02	0.03	0.10	0.06	0.18	0.02
fruit	0.01	0.07	0.04	0.09	0.06	0.29	0.06	0.06	0.05	0.03	0.07	0.03	0.12	0.04
.	0.01	0.05	0.06	0.09	0.07	0.26	0.06	0.06	0.06	0.02	0.13	0.02	0.11	0.01

(a) Standard Transformer.

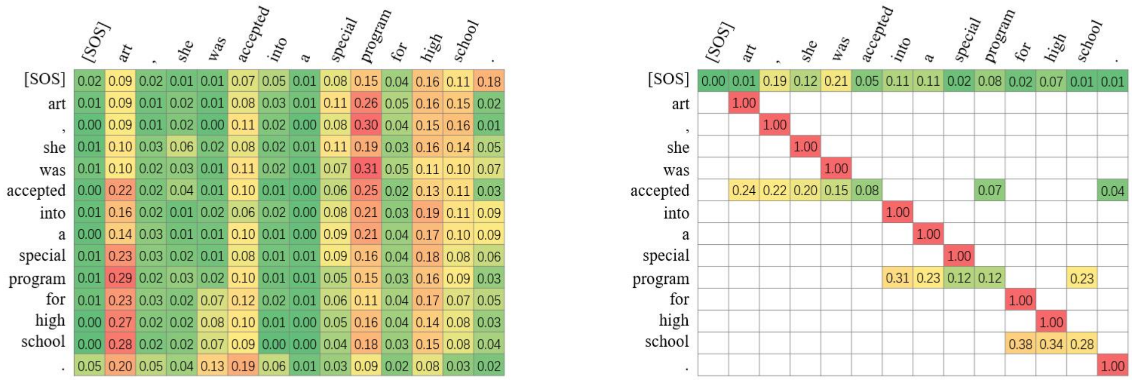
	[SOS]	hello	,	i	am	going	to	the	store	to	buy	some	fruit	.
[SOS]	0.00	0.05	0.08	0.00	0.02	0.08	0.04	0.20	0.01	0.04	0.01	0.33	0.07	0.09
hello		1.00												
,			1.00											
i				1.00										
am					1.00									
going		0.08	0.08	0.17	0.13	0.04			0.26		0.17			0.08
to							1.00							
the								1.00						
store							0.30	0.23	0.47					
to										1.00				
buy										0.20	0.26		0.55	
some												1.00		
fruit												0.55	0.45	
.														1.00

(b) Transformer+DRA.

Figure 6: Attention weights visualization of example 1

Example 2	
Speaker1:	My niece is super talented lately.
Speaker2:	What is her best talent?
Speaker1:	Art, she was accepted into a special program for high school.
Gold Resp:	Does she draw or paint? How many students are in this program?
Transformer:	Wow, that is a pretty cool name.
Transformer+DRA:	Oh wow, that is impressive.
DialogBERT:	That's great. What kind of job?
DialogBERT+DRA:	Wow, that is a big accomplishment.

Table 6: Example responses from different models.



	[SOS]	art	,	she	was	accepted	into	a	special	program	for	high	school	.
[SOS]	0.02	0.09	0.02	0.01	0.01	0.07	0.05	0.01	0.08	0.15	0.04	0.16	0.11	0.18
art	0.01	0.09	0.01	0.02	0.01	0.08	0.03	0.01	0.11	0.26	0.05	0.16	0.15	0.02
,	0.00	0.09	0.01	0.02	0.00	0.11	0.02	0.00	0.08	0.30	0.04	0.15	0.16	0.01
she	0.01	0.10	0.03	0.06	0.02	0.08	0.02	0.01	0.11	0.19	0.03	0.16	0.14	0.05
was	0.01	0.10	0.02	0.03	0.01	0.11	0.02	0.01	0.07	0.31	0.05	0.11	0.10	0.07
accepted	0.00	0.22	0.02	0.04	0.01	0.10	0.01	0.00	0.06	0.25	0.02	0.13	0.11	0.03
into	0.01	0.16	0.02	0.01	0.02	0.06	0.02	0.00	0.08	0.21	0.03	0.19	0.11	0.09
a	0.00	0.14	0.03	0.01	0.01	0.10	0.01	0.00	0.09	0.21	0.04	0.17	0.10	0.09
special	0.01	0.23	0.03	0.02	0.01	0.08	0.01	0.01	0.09	0.16	0.04	0.18	0.08	0.06
program	0.01	0.29	0.02	0.03	0.02	0.10	0.01	0.01	0.05	0.15	0.03	0.16	0.09	0.03
for	0.01	0.23	0.03	0.02	0.07	0.12	0.02	0.01	0.06	0.11	0.04	0.17	0.07	0.05
high	0.00	0.27	0.02	0.02	0.08	0.10	0.01	0.00	0.05	0.16	0.04	0.14	0.08	0.03
school	0.00	0.28	0.02	0.02	0.07	0.09	0.00	0.00	0.04	0.18	0.03	0.15	0.08	0.04
.	0.05	0.20	0.05	0.04	0.13	0.19	0.06	0.01	0.03	0.09	0.02	0.08	0.03	0.02

(a) Standard Transformer.

	[SOS]	art	,	she	was	accepted	into	a	special	program	for	high	school	.
[SOS]	0.00	0.01	0.19	0.12	0.21	0.05	0.11	0.11	0.02	0.08	0.02	0.07	0.01	0.01
art		1.00												
,			1.00											
she				1.00										
was					1.00									
accepted		0.24	0.22	0.20	0.15	0.08				0.07				0.04
into							1.00							
a								1.00						
special									1.00					
program							0.31	0.23	0.12	0.12			0.23	
for											1.00			
high												1.00		
school											0.38	0.34	0.28	
.														1.00

(b) Transformer+DRA.

Figure 7: Attention weights visualization of example 2