Incorporate Directed Dependency Relation Graph into Transformer Block for Multi-turn Dialogue Generation

Anonymous ACL submission

Abstract

Because of the compositionality of natural language, syntactic structure is a key factor for semantic understanding in dialogue generation tasks. However, the widely adopted Transformer is hard to learn the compositionaity effectively, because the position embeddings contain less semantic relation information. To explicit model the compositionaity of language, we limit the information flow between words by constructing directed dependency relation graph and propose Dependency Relation Attention (DRA) to replace position embeddings. Experimental results demonstrate that DRA can further improve the performance of state-of-the-art models for multi-turn dialogue generation.

1 Introduction

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) for dialogue generation tasks to extract context features(Su et al., 2019; Liu et al., 2020; Song et al., 2021). The standard Transformer block consists of a multi-head attention network and a feed-forward neural network followed by residual connections(He et al., 2016) and normalization. Since there is no recurrence and no convolution, the network simply adds the position embeddings to the corresponding word embeddings to make use of the order of sequence.

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". 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).

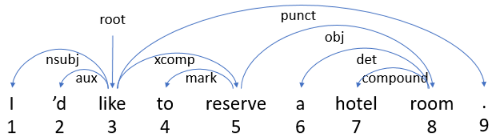


Figure 1: An example of dependency parse.

This demonstrates that syntactic structure plays an important role in NLP. However, the Transformer block contains no explicit modeling of syntax, and we believe that the following reasons make it difficult for the Transformer block to learn syntactic structure in the training of dialogue generation: (1) The Transformer encoder learns the local position information that can only be effective in masked language modeling(Wang and Chen, 2020). (2) The computation of attention weights on unrelated word pairs is redundant and decreases performance.

To obtain better distributed representations of utterances, in this paper, we propose Dependency Relation Attention to incorporate dependency relation knowledge that contains syntactic structure information into the Transformer block. Specifically, as shown in Figure 1, we use the dependency parser(Chen and Manning, 2014) in the Stanford-CoreNLP toolkit(Manning et al., 2014) to build directed dependency relation graph. Then, the Dependency Relation Mask is generated to avoid performing attention on words without dependency relations. The fusion of information among words depends on the direction specified by the dependency relation. 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 multi-turn dialogue generation task by conducting experiments on two datasets.

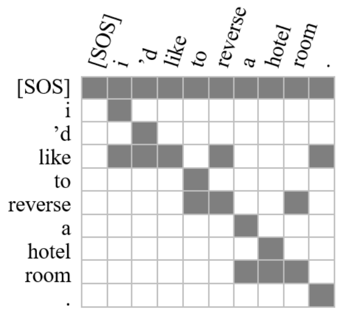


Figure 2: Dependency Relation Mask.

2 Method

In multi-turn dialogue generation task, 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 $Y = \{y_1, ..., y_t\}$:

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

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

2.1 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:

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

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:

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

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

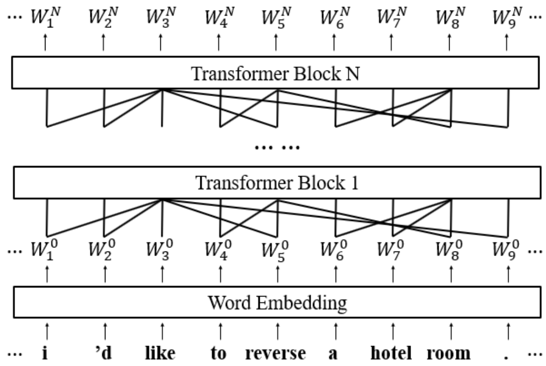


Figure 3: Illustration of applying DRA to standard Transformer encoder. Dependency Relation Mask is used to model the semantic relationship between words instead of position embeddings.

2.2 Dependency Relation Attention

The main idea of our proposed method is to use Dependency Relation Attention (DRA) to model the relationships between words instead of 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:

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

3 Experiments

Our method aims to further enhance the semantic understanding of the Transformer encoder. It can be applied to models that use Transformer blocks to map context into feature vectors. In this section, we explore whether our method is effective.

Model	DailyDialog			EmpatheticDialogues		
	PPL	BLEU-2	Dist-2	PPL	BLEU-2	Dist-2
HRED	37.005	17.865	2.180	45.399	13.741	2.037
HRAN	28.411	18.359	8.073	40.901	19.002	4.355
ReCoSa	20.799	21.354	19.137	35.289	19.638	**8.878**
Transformer	19.168	19.314	18.317	33.052	18.643	8.222
Transformer+DRA	**18.682**	20.822	19.358	**32.209**	20.488	8.503
DialogBERT	20.766	18.008	16.370	36.325	19.404	6.356
DialogBERT+DRA	19.279	**21.744**	**19.519**	33.386	**21.247**	8.687

Table 1: Automatic evaluation results on DailyDialog and EmpatheticDialogues. The best results are in bold.

3.1 Settings

3.1.1 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. Words with word frequencies less than 3 are replaced by "[UNK]". The length of dialogue turns and the utterance length are limited to 4 and 50, respectively.

3.1.2 Compared Methods

We apply DRA to Transformer(Vaswani et al., 2017) and DialogBERT(Gu et al., 2021), and compare the performance before and after the modification. In addition, the following methods are compared: HRED(Serban et al., 2016), HRAN(Xing et al., 2018), and ReCoSa(Zhang et al., 2019).

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. We initialize the word embedding layers with GloVe 300-dimensional word embeddings(Pennington et al., 2014). The batch size is 40. All models are trained by the AdamW(Loshchilov and Hutter, 2018) optimizer with an initial learning rate of 5e-4.

3.1.3 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).

Model	+2	+1	+0	Avg.
HRED	3.7	45.3	51.0	0.53
HRAN	26.7	62.7	10.7	1.16
ReCoSa	37.7	52.3	10.0	1.28
Transformer	40.3	56.0	3.7	1.37
Transformer+DRA	45.7	48.3	6.0	1.40
DialogBERT	24.3	69.7	6.0	1.18
DialogBERT+DRA	46.3	49.0	4.7	**1.42**

Table 2: Human evaluation results. (in %)

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).

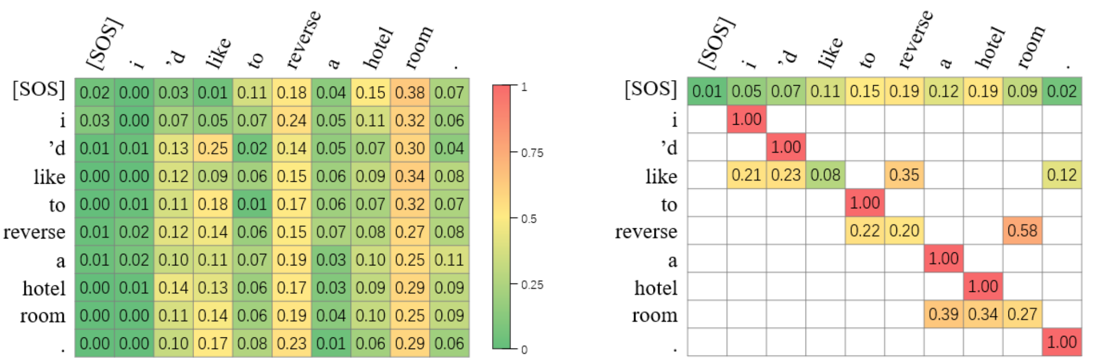
3.2 Experimental Results

Table 1 gives the automatic evaluation results. For both datasets, Transformer+DRA and DialogBERT+DRA achieved the best performance on PPL and BLEU-2 respectively. DialogBERT+DRA achieved comparable Dist-2 scores in contrast to ReCoSa. 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 Table 2. The Fleiss' kappa score(Fleiss, 1971) for

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?
HRED:	That's great!
HRAN:	I'm sure he is going to be a great time.
ReCoSa:	That's really great. What kind of her does she do?
Transformer:	Wow, that is a pretty cool name.
Transformer+DRA:	Oh wow! That is impressive. I bet she is proud of her.
DialogBERT:	That's great. What kind of job?
DialogBERT+DRA:	Wow, that is impressive. You must be so proud.

Table 3: Example responses from different models.



(a) Standard Transformer. (b) Transformer+DRA.

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

assessing agreement among annotators was 0.510, 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.

3.3 Discussions

Table 3 is an example of a generated dialogue that demonstrates that Dependency Relation Attention can help Transformer and DialogBERT generate better responses.

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 models. 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, attention weights are assigned to appropriate parts for each word. For example, when updating the hidden state of "reverse", 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.

4 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 multi-turn 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.

References

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

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

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.

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

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.

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.

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.

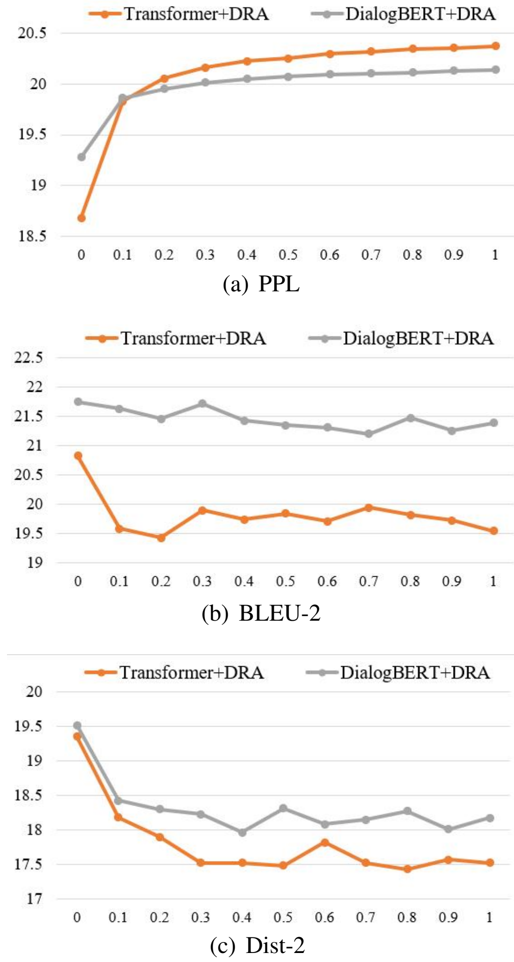
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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) PPL

(b) BLEU-2

(c) Dist-2

Figure 5: The result of parsing errors.

A 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, that is, mask the attention weights with dependency relations and those without dependency relations will not be masked. Figure 5 show how the parsing errors affect PPL, BLEU-2, Dist-2 of models on DailyDialog test set. The horizontal axis in the figure represents the proportion of parsing errors. It show that our proposed method has certain robustness, especially for the hierarchical Transformer encoder architecture.

B Comparison of running time

Dependency relation parsing takes additional computations and the running time of the proposed approach and traditional baselines are compared. We show the average time taken by different models to

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	0.023s	0.148s

Table 4: Comparison of running time.

generating response for each dialogue in DailyDialog in Table 4 (*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.