Bi-Directional Multi-Granularity Generation Framework for Knowledge Graph-to-Text with Large Language Model

Haowei Du [1,2,3], Chen Li [3], Dinghao Zhang[3], Dongyan Zhao[1,2,*]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]State Key Laboratory of Media Convergence Production Technology and Systems [3]Ant Group
2301112050@stu.pku.edu.cn, wenyou.lc@antgroup.com,
zhangdinghao.zdh@antgroup.com, zhaodongyan@pku.edu.cn

Abstract

The knowledge graph-to-text (KG-to-text) generation task aims to synthesize coherent and engaging sentences that accurately convey the complex information derived from an input knowledge graph. Existing methods generate the whole target text based on all KG triples at once and may incorporate incorrect KG triples for each sentence. To this end, we propose the bi-directional multi-granularity generation framework. Instead of generating the whole text at a time, we construct the sentence-level generation based on the corresponding triples and generate the graph-level text as a result. Moreover, we design a backward relation-extraction task to enhance the correctness of relational information. Our method achieves the new state-of-the-art in benchmark dataset WebNLG and further analysis shows the efficiency of different modules.

1 Introduction

Knowledge graph (KG) is a structured data representation form that contains rich knowledge information and is more convenient for processes such as information retrieval and reasoning. Although KGs facilitate computational processes, it is difficult for humans to intuitively understand the content in KGs, so the proposed KG-to-text generation task aims to produce correct descriptive text for the input KG. KG-to-text has various applications, like question-and-answer (Pal et al., 2019) and dialogue systems (Zhou et al., 2018). Moreover, with the population of large language models (LLM), KG-to-text plays an important role in transforming structured knowledge into texts to alleviate hallucination in LLMs (Ji et al., 2023).

Recent works insert extra graph modules into pretrained language model (PTM) and decode the whole target text based on all KG triples in one round (Ke et al., 2021; Zhao et al., 2023). With the size of KG growing, the full generation enlarges and there are multiple sentences to describe the KG with different sentences describing different aspects. However, the model may incorporate incorrect KG triples to generate the current sentence, which undermines the overall generation.

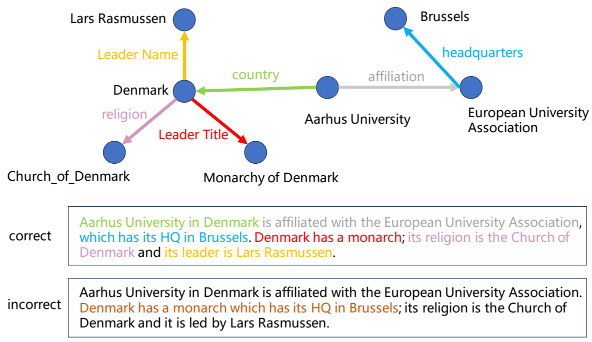

Figure 1: One example from WebNLG dataset. There are 6 triples in this KG to generate the text: <Denmark ,Leader Title, Monarchy of Denmark>; <Denmark, religion, Church of Denmark>; <Denmark, Leader Name, Lars Rasmussen>; <Aarhus University, country, Denmark>; <Aarhus University, affiliation, European University Association>; <European University Association, headquarters, Brussels>. The "incorrect" denotes the incorrect generation of baseline model.

We take an example in WebNLG dataset (Gardent et al., 2017) in Figure 1. There are 6 triples in the KG and the target generation contains two sentences: "Aarhus University in Denmark is affiliated with the European University Association, which has its HQ in Brussels." and "Denmark has a monarch; its religion is the Church of Denmark and its leader is Lars Rasmussen.". The first sentence describes "Aarhus University" and its affiliation "European University Association". The second sentence describes the political and religious information of "Denmark", so it should be generated based on the 3 triples including "<Denmark ,Leader Title, Monarchy of Denmark>"; "<Denmark, religion, Church of Denmark>"; "<Denmark, Leader

*Corresponding Author

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 147–152
August 11-16, 2024

Name, Lars Rasmussen>". The baseline model misunderstands the triple "<Aarhus University, affiliation, European University Association>" for this sentence and generates the incorrect text.

To enhance the fine-grained information of each sentence generated by the model, we propose our bi-directional multi-granularity generation framework (BDMG). Instead of generating the whole text at a time, we construct the sentence-level generation based on the corresponding triples and generate the graph-level text as a result. First, We prompt the model to find the subset of triples in KG which are needed for the current sentence. Then the model generate the current text based on these triples. Finally the model aggregates the sentence-level generation into the final result. Moreover, we design a backward relation-extraction (RE) task to enhance the correctness of relational information. Specifically, we randomly choose a number of triples in KG and ask the model to infer the relations between the head and tail entities. The model is jointly optimized by the two tasks.

We conduct experiments on the benchmark dataset in KG-to-Text task, WebNLG, and derives the new state-of-the-art (SOTA), which shows the efficiency of our bi-directional multi-granularity generation framework. Further experiments demonstrate the importance of step by step sentence-level generation and backward relation extraction to the KB-to-Text task.

We conclude our contributions as follows: **1**. We propose the bi-directional multi-granularity generation framework, where the model generates the sentence-level information at first and aggregate into generating the KG-level text. **2**. We design the backward relation extraction task into enhancing the relational information of triples in KG, which improves the overall performance of generating text from KG triples. **3**. We conduct experiments on the benchmark dataset WebNLG and achieves the new SOTA.

2 Related Work

2.1 KG-to-Text

To capture the KG structural information, many recent works on KG-to-text generation encode the graph structure directly using graph neural networks (GNNs) (Guo et al., 2019; Zhao et al., 2020; Ribeiro et al., 2020; Li et al., 2021) or graph-transformers (Schmitt et al., 2020) and then decode into texts. DUALENC (Zhao et al., 2020) feeds the input KG into two GNN encoders for order planning and sentence generation. Graformer (Schmitt et al., 2020) introduces a model that combines relative position information to compute self-attention. Other approaches (Wang et al., 2021; Liu et al., 2022; Guo et al., 2019; Ribeiro et al., 2020) first linearize KG into sequences and then feed them into the sequence-to-sequence (Seq2Seq) model for generating desired texts. Existing works (Zhao et al., 2020) have shown that the linearized order of the given triples has an effect on the quality of generated text. Previous works mainly use graph traversal (Li et al., 2021) or multistep prediction (Su et al., 2021) methods for triple order generation. Li et al. (2021) uses the relation-biased BFS (RBFS) strategy to traverse and linearize KGs into sequences. Zhao et al. (2020) uses the content planner to select one of the remaining unvisited triples at each step until all triples have been visited.

Recent KBQA methods (Du et al., 2022, 2023a) employ GNN to solve queries based on the KB, which is hard to transfer to LLM because of large computation cost. However, KG-to-text task bridges the gap between KG and LLM. KG can be converted to natural text and then apply the LLM to solve the query. Moreover using query rewritten methods (Du et al., 2023b), multi-turn KG-based queries can be refined into semantic-complete query and answered by LLM based on the natural text generated from KB triples by KG-to-text methods.

2.2 Chain of Thought

Recent works on CoT prompting is prompting LLMs step by step to leverage their comprehension and reasoning abilities to answer questions. Zero-shot-CoT (Kojima et al., 2022) adopts a two stage design, which requires LLMs to first generate intermediate rationale and then produce an answer. Wang et al. (2022) introduced iCAP, which iteratively prompts a fine-tuned small-scale LLM to generate CoTs and then combines the generated rationales to formulate answers. Least-to-Most (Zhou et al., 2022) requires LLMs to first decompose a complex question into sub-questions and then sequentially solve them to arrive at the final answer.

3 Methodology

In this part, first we introduce the task of KG-to-Text, then we introduce our BDMG approach. Our

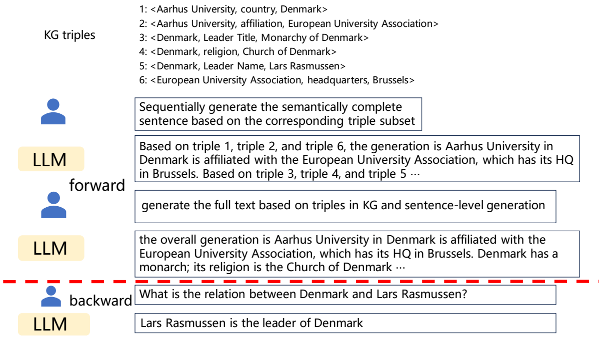

Figure 2: Pipeline of our approach BDMG. It includes forward sequential sentence-level generation and backward relation extraction.

method includes two modules: forward sequential sentence-level generation and backward relation extraction. The forward generation process absorbs the thought of **Divide-and-Conquer** algorithm (Smith, 1985). We ask the LLM to decide the triple subset which should be generated in current sentence, and merge the generation result of different subsets into the full generation of KG.

3.1 Task formulation

The aim is to generate accurate text to describe the input KG. The input KG consists of some triples and $\mathcal{G} = \{\langle h, r, t\rangle \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ and $\mathcal{R}$ are sets of entities and relations, respectively. Following (Ke et al., 2021), we linearize the input KG as $\mathcal{G}_{\text{linear}} = (w_1, w_2, \cdots, w_m)$, where m is the number of tokens. The target is to generate the text $\mathcal{T} = (t_1, t_2, \cdots, t_n)$, which gives an accurate and complete description of the information in the input KG.

3.2 Forward Sentence-Level Generation

In this part, we decompose the generation of the text to describe the full KG into a sequential decoding problem: the model sequentially generate a semantically complete sentence with the sentence-specific subset of KG triples. Then the model generates the full text of KG based on the triples and the sentence-level generation. The generation process can be formulated as follows:

$$
\begin{aligned}
&P(\mathbf{cot}, T|\mathbf{KG}) \\
&= P((s_1,t_1),\cdots,(s_n,t_n),T|\mathbf{KG}) \\
&= \prod_{i=1}^{n} P((s_i,t_i)|(s_1,t_1),\cdots,(s_{i-1},t_{i-1}),\mathbf{KG})\cdot \\
&P(T|(s_1,t_1),\cdots,(s_n,t_n),\mathbf{KG}) \\
&= \prod_{i=1}^{n} P(t_i|(s_1,t_1),\cdots,(s_{i-1},t_{i-1}),\mathbf{KG})\cdot \\
&\prod_{i=1}^{n} P(s_i)|t_i,(s_1,t_1),\cdots,(s_{i-1},t_{i-1}),\mathbf{KG})\cdot \\
&P(T|(s_1,t_1),\cdots,(s_n,t_n),\mathbf{KG})
\end{aligned}
$$

where $\mathbf{cot}$ denotes the sequential sentence-level generation, t_i denotes the i-th sentence-specific triple subset in the original KG, s_i denotes the sentence-level generation based on this triple subset, n denotes the sentence number, T denotes the overall text generation with the full KG triples.

In the example in Figure 2 , There are two semantically complete sentences in the target text, i.e. $n = 2$. The first sentence s_1 is “Aarhus University in Denmark is affiliated with the European University Association, which has its HQ in Brussels”, which describes the “Aarhus University” and its affiliation. The triplet subset corresponding to this sentence t_1 is ”<Aarhus University, country, Denmark>; <Aarhus University, affiliation, European University Association>; <European University Association, headquarters, Brussels>”. The second sentence s_2 describes the entity “Denmark”.

The cross-entropy loss is utilized to optimize the model:

$$
\begin{aligned}
&L_{seq} = -\mathbf{log}P((s_1,t_1),\cdots,(s_n,t_n),T|\mathbf{KG}) \\
&= -\sum_{i=1}^{n}\mathbf{log}P(t_i|(s_1,t_1),\cdots,(s_{i-1},t_{i-1}), \\
&\mathbf{KG}) - \sum_{i=1}^{n}\mathbf{log}P(s_i)|t_i,(s_1,t_1),\cdots,(s_{i-1}, \\
&t_{i-1}),\mathbf{KG}) - \sum_{i=1}^{n}\mathbf{log}P(T|(s_1,t_1),\cdots,(s_n, \\
&t_n),\mathbf{KG})
\end{aligned}
$$

3.3 Backward Relation Extraction

To help the model capture the correct relational information between the head and tail entities, we design the backward relation extraction task. Specifically, we randomly sample a number of triples from the KG and prompt the model to infer the relation between its head and tail entities based on the text generation of the KG. Such as the triple " <European University Association, headquarters, Brussels>", we prompt the model as "what is the relation between European University Association and Brussels based on the text $\cdots$", and the target answer is "The headquarters of European University Association are in Brussels". The objective function is as follows:

$$L_{re} = -\mathbf{log}P(r|h,t,T) = -\mathbf{log}\prod_{i=1}^{m} P(r_i|r_{<i},h,t,T)$$

where h, t, r denotes the head entity, tail entity and relation of the sampled triple, T denotes the generated text to describe the KG, and m denotes the answer length.

3.4 Training and Inference

Our model is jointly optimized by the sequential sentence-level generation loss and the backward RE loss:

$$L = \alpha_1 L_{seq} + \alpha_2 L_{re}$$

where α_1 and α_2 are parameters to tune. In the training of sentence-level generation, we add special tokens, "[SEQ]" and "[RES]" before the sentence-level generation and the final aggregated text of the full KG. In the inference stage, we take the text after the "[RES]" token as the final result.

4 Experiments

4.1 Dataset and Backbone

WebNLG (Gardent et al., 2017) is a frequently used benchmark dataset in KG-to-Text task. A sample in the dataset contains one to seven triples. The text to describe the KGs mostly contains multiple sentences, which is appropriate for our sequential sentence-level generation. We followed the existing work (Ke et al., 2021) to use the more challenging split (Constrained) version of 2.0 (Shimorina and Gardent, 2018), which guarantees that there is no overlap on triples of input graphs among train / validation / test set. We utilize the widespread LLM Flan T5 (Chung et al., 2022) with sizes from 3B to 11B as the backbone model.

Models	BLEU	METROE	ROUGE
SOTA-NPT	48.00	36.00	65.00
KGPT	59.11	41.20	69.47
JointGT	61.01	46.32	73.57
Plan Selection	62.12	46.78	73.96
Flan T5 3B	67.56	47.67	78.10
BDMG 3B	68.75	48.90	79.58
Flan T5 11B	69.32	49.22	79.89
BDMG 11B	**70.65**	**50.30**	**81.36**

Table 1: Experimental results on WebNLG dataset. We conduct 5 experiments with different random seeds and our method significantly beats the prior SOTA Plan-Selection, with p-value less than 0.001.

4.2 Implementation Details

To reduce memory cost and preserve prior knowledge, we adopt LORA adapter (Zhang et al., 2023) to the LLM and freeze original parameters. The number of trainable parameters of BDMG-3B is 3M, only 0.1% of total parameters. We set the LoRA rank and scaling factor to 8 and 16. The training batch size is set to 4 for BDMG-3B and 2 for BDMG-11B. We utilize AdamW as the optimizer and the initial learning rate is set to 3e-5. The value of hyper-parameter α_1 and α_2 in section 3.4 is set to 1.0 and 0.6. We make use of off-shelf NLP tools spaCy (Vasiliev, 2020) to link the entity in KG to the annotated text which describes the full KG, thus construct the target of sentence-level generation. Following (Ke et al., 2021) we utilize METEOR (Banerjee and Lavie, 2005), ROUGEL (Lin, 2004) and BLEU-4 (Papineni et al., 2002) as evaluation metrics. We compare our methods with existing methods including SOTA-NPT (Ke et al., 2019), KGPT (Chen et al., 2020), JointGT (Ke et al., 2021) and Plan Selection (Zhao et al., 2023)

4.3 Results

In Table 1, our approach BDMG-11B beats the prior SOTA, Plan Selection, with about 8.5 BLEU, 3.6 METEOR, 7.4 ROUGE score. Compared with the backbone Flan T5, our model outperforms by about 1.2 BLEU, 1.2 MeTEOR and 1.5 ROUGE score with 3B version, as well as 1.3 BLEU, 1.1 METEOR, 1.5 ROUGE score with 11B version. It demonstrate the efficiency of bi-directional multi-granularity generation framework, including forward sequential sentence-level generation and backward relation extraction.

Models	BLEU	METROE	ROUGE
- COT	68.03	48.12	78.55
- RE	68.45	48.56	79.14
BDMG	68.75	48.90	79.58

Table 2: Ablation results with Flan T5 3B as backbone. **- COT** denotes removing the sequential sentence-level generation and directly generate the final text to describe the full KG, **- RE** denotes removing the backward relation extraction task.

4.4 Ablation

In Table 2, we conduct ablation experiments to evaluate different modules of our method. By removing the sequential sentence-level generation, the performance drops by about 0.7 BLEU, 0.8 METEOR and 1.0 ROUGE. It shows the importance of choosing triple subset from the full KG to generate the semantically complete sentence sequentially. By removing the backward RE task, the model drops by 0.3 BLEU, 0.3 METREOR and 0.4 ROUGE. It shows the backward RE task enhances the relational information between KG entities for model and improves the overall generation.

5 Conclusion

In this paper, we propose our bi-directional multi-granularity generation framework. Instead of generating the whole text at a time, we construct the sentence-level generation based on the corresponding triples and generate the graph-level text as a result. We conduct experiments on benchmark dataset and significantly achieves the new SOTA. Further analysis shows the efficiency of different modules. This work was completed by the first author during internship in Ant Group.

Limitations

We propose our bi-directional multi-granularity generation framework and demonstrate our efficiency on the benchmark dataset WebNLG. Our method focuses on the sequential sentence-level generation, which applies to larger KG with multiple sentences as description, and do not apply to simple KG with only one sentence.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No.2021YFC3340303).

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. Kgpt: Knowledge-grounded pretraining for data-to-text generation. *arXiv preprint arXiv:2010.02307*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Haowei Du, Quzhe Huang, Chen Li, Chen Zhang, Yang Li, and Dongyan Zhao. 2023a. Relation-aware question answering for heterogeneous knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13582–13592.

Haowei Du, Quzhe Huang, Chen Zhang, and Dongyan Zhao. 2022. Knowledge-enhanced iterative instruction generation and reasoning for knowledge base question answering. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 431–444. Springer.

Haowei Du, Dinghao Zhang, Chen Li, Yang Li, and Dongyan Zhao. 2023b. Multi-granularity information interaction framework for incomplete utterance rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2576–2581.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. 2019. Sentilare: Sentiment-aware language representation learning with linguistic knowledge. *arXiv preprint arXiv:1911.02493*.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang.

2021. Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. *arXiv preprint arXiv:2106.10502*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. *arXiv preprint arXiv:2106.01623*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jin Liu, Chongfeng Fan, Fengyu Zhou, and Huijuan Xu. 2022. Syntax controlled knowledge graph-to-text generation with order and semantic consistency. *arXiv preprint arXiv:2207.00719*.

Vaishali Pal, Manish Shrivastava, and Irshad Bhat. 2019. Answering naturally: Factoid to full length answer generation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 1–9.

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.

Leonardo FR Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling global and local node contexts for text generation from knowledge graphs. *Transactions of the Association for Computational Linguistics*, 8:589–604.

Martin Schmitt, Leonardo FR Ribeiro, Philipp Dufter, Iryna Gurevych, and Hinrich Schütze. 2020. Modeling graph structure via relative position for text generation from knowledge graphs. *arXiv preprint arXiv:2006.09242*.

Anastasia Shimorina and Claire Gardent. 2018. Handling rare items in data-to-text generation. In *Proceedings of the 11th international conference on natural language generation*, pages 360–370.

Douglas R Smith. 1985. The design of divide and conquer algorithms. *Science of Computer Programming*, 5:37–58.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. *arXiv preprint arXiv:2108.13740*.

Yuli Vasiliev. 2020. *Natural language processing with Python and spaCy: A practical introduction*. No Starch Press.

Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively prompt pre-trained language models for chain of thought. *arXiv preprint arXiv:2203.08383*.

Qingyun Wang, Semih Yavuz, Victoria Lin, Heng Ji, and Nazneen Rajani. 2021. Stage-wise fine-tuning for graph-to-text generation. *arXiv preprint arXiv:2105.08021*.

Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2481–2491.

Feng Zhao, Hongzhi Zou, and Cheng Yan. 2023. Structure-aware knowledge graph-to-text generation with planning selection and similarity distinction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8693–8703.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.

A Example Appendix

This is an appendix.