Enhancing Graph-to-Text Systems in Low-Resource Settings: Distilling Chain-Of-Thought Reasoning For Task-Specific Workflows

David Guzman Piedrahita[*]
Department of Informatics
University of Zürich
Rämistrasse 71, CH-8006 Zürich, Switzerland
david.guzmanpiedrahita@uzh.ch

Arnisa Fazla[†]
Department of Informatics
University of Zürich
Rämistrasse 71, CH-8006 Zürich, Switzerland
arnisa.fazla@uzh.ch

Anna Kiepura[‡]
Department of Information Technology and Electrical Engineering
ETH Zürich
Gloriastrasse 35, 8092 Zürich, Switzerland
akiepura@ethz.ch

Abstract

Knowledge graphs are essential for organizing vast amounts of information, yet their structured nature can be challenging for non-experts to interpret directly. Graph-based text generation addresses this issue by converting graph data into natural language, facilitating user understanding. While recent advancements in Large Language Models (LLMs) have shown promise in this task, their high resource consumption limits their feasibility. This study proposes a pipeline of smaller language models (SLMs) that distill reasoning capabilities from external LLMs, specifically GPT-3.5 Turbo, and evaluates their performance on the graph-based text generation task using the WebNLG dataset. By augmenting the dataset with intermediate reasoning steps, we fine-tune two models in the pipeline: Triples-to-Reasoning and Reasoning-to-Text. Our results indicate that the pipeline consisting of FLAN-T5-base models outperforms the baseline single FLAN-T5-base model approach, showcasing the effectiveness of intermediate reasoning, while the FLAN-T5-small model did not yield similar improvements, emphasizing the importance of model capacity. This work highlights the potential for SLM pipelines to emulate task decomposition and step-by-step reasoning, offering a pathway for deploying efficient and interpretable models in low-resource environments [4].

1 Introduction

Knowledge graphs are commonly used in AI systems such as recommender systems, question-answering, and information retrieval [19]. However, they are often difficult for non-experts to interpret due to their structured nature. Graph-based text generation addresses this challenge by converting graph data into natural language, making complex information more accessible. While

[*]Corresponding author. All authors contributed equally.
[†]All authors contributed equally.
[‡]All authors contributed equally.
[4]Code is available at: https://github.com/davidguzmanp/Graph-to-Text-LLM-with-dataset-augmentation

Latinx in AI @ NeurIPS-24 (LXAI 2024).

recent advancements in Large Language Models (LLMs) have shown promising results for graph-based text generation [34], their high resource consumption limits their practical use.

Chain-of-Thought (CoT) prompting has been shown to enhance LLM performance by encouraging step-by-step reasoning [30], but its applicability to smaller models is uncertain. Recent work has focused on fine-tuning Small Language Models (SLMs) to distill reasoning capabilities from LLMs, improving their performance on tasks like mathematical and table-based reasoning [36, 15, 33]. However, these SLMs still struggle with complex reasoning compared to larger models.

In this paper, we propose a pipeline using SLMs to distill reasoning from GPT-3.5 Turbo [22] and evaluate its effectiveness on graph-based text generation using the WebNLG dataset [16]. Our pipeline improves the performance of SLMs by incorporating intermediate reasoning steps, offering a potential solution for deploying reasoning capabilities in resource-constrained environments like edge computing.

2 Background

Early methods for incorporating Knowledge Graphs (KGs) into neural models used text-to-text frameworks, converting KG triples into sequences for natural language generation [14, 28, 17, 3]. These approaches focused on preserving graph structure using rule-based techniques [10, 32, 9] and later moved to graph-specific models like Graph Convolutions (GCNs) [12, 23, 5, 24] and Graph Neural Networks (GNNs), which integrated attention mechanisms for direct graph input [7, 13].

However, studies show that linearizing graph structures can damage the graph's connectivity and hinder knowledge transfer during fine-tuning [27, 1, 11, 26]. To address these, methods like structure-aware semantic aggregation [11] and graph-aware adapters [26] were developed to better align graph and text representations.

Despite progress in graph-to-text generation, challenges remain in scalability, interpretability, and over-smoothing [31]. Large language models (LLMs) have the potential to address these issues, with recent research highlighting the importance of model size for multi-step reasoning and knowledge transfer Wei et al.. Additionally, knowledge distillation offers a way to transfer capabilities from larger to smaller models, making them more efficient for resource-constrained tasks [6, 8, 18].

3 Methodology

In this section, we explain our pipelined approach to the graph-based text generation task, which consists of four key stages:

Graph Structure to Triples Preprocessing We convert input graphs into a linearized sequence of triples in the format `<H>Head <R>Relation <T>Tail`. Special tokens for head, relation, and tail entities are added to the tokenizer, and samples with missing target texts are removed to maintain data quality.

Pipeline Approach with Intermediate Reasoning We leverage GPT3.5-turbo to generate reasoning sentences from triples, creating a two-stage pipeline. In this process, we generate 1,000 samples with ChatGPT, of which 700 are designated for training, 100 for validation, and 200 for testing. The first model, Triples-to-Reasoning, converts triples into coherent, yet separate sentences. The second model, Reasoning-to-Text, transforms these sentences into a single natural language description. During inference, the pipeline sequentially processes graph triples through both models, converting structured data into coherent natural language descriptions. For example:

```
<H>Marie Curie<R>BornIn<T>Warsaw
<H>Marie Curie<R>Field<T>Chemistry
```

The output from the Triples-to-Reasoning model is composed of the following sentences:

```
Marie Curie was born in Warsaw.
Marie Curie specialized in the field of Chemistry.
```

The Reasoning-to-Text model then produces the final output:

```
Marie Curie, born in Warsaw, was a scientist known for her research in Chemistry.
```

This approach aims to enhance smaller models' ability to remain faithful to the input graph and reduce hallucinations, ensuring the generated text aligns more accurately with the original data.

4 Experiment Settings

Our experiments utilize the WebNLG dataset [16],[5] consisting of triples that represent entities and their relationships, paired with corresponding natural language descriptions. We employ two variants of the FLAN-T5 model [4] in these experiments. The first is FLAN-T5-small (60M parameters), selected for its efficiency in low-resource settings. The second is FLAN-T5-base (220M parameters), used to evaluate performance scalability with increased model capacity. Both models are fine-tuned for the graph-to-text generation task, according to the methodology laid out in Section 3. Model performance is assessed using several metrics. BLEU [21] measures the overlap of n-grams between the generated text and reference text. CHR F++ [20] evaluates character-level n-gram precision, recall, and F-score to capture finer-grained similarities. BERT-Scores [35] assesses semantic similarity between generated and reference texts using contextual embeddings. These metrics together provide a comprehensive evaluation of both lexical and semantic alignment with the reference data.

For baseline comparisons, we adapt the methodology of Ribeiro et al. [25], which involves task-adaptive pre-training and fine-tuning transformer models for graph-to-text tasks. A pre-trained FLAN-T5-small model is fine-tuned on the WebNLG dataset to establish performance benchmarks.

For hyperparameter settings, during task-adaptive pre-training, we train the model for 30 epochs with a masking probability of 0.15, using the AdamW optimizer with a learning rate of 1×10^{-4} and with an early stopping patience of 3 epochs. In the fine-tuning phase, we fine-tune for 5 epochs with a batch size of 16 and maintain the learning rate at 1×10^{-4}, again using early stopping but with a patience of 2 epochs.

5 Results

Table 1: Metric scores of T5-flan small and base models on the 2020 WebNLG test set. Single model results (above midrule) vs. pipeline results (below). The pipeline improves performance for T5-flan-base but not for T5-flan-small.

Method	BERT precision	BLEU	chrF++
T5-flan-small Single Model	0.916	28.3	0.54
T5-flan-base Single Model	0.929	34.3	0.6
T5-flan-small Pipelined Model	0.897	19.98	0.5
T5-flan-base Pipelined Model	**0.935**	**37.46**	**0.6**

Table 2: Metric scores of pipeline methods (triples-to-reasoning and reasoning-to-text) evaluated on 200 test samples containing intermediate reasoning data for separate model evaluation.

Test Dataset	BERT precision	BLEU	chrF++
T5-flan-small Triples-to-Reasoning	0.969	70.98	0.81
T5-flan-small Reasoning-to-Text	0.929	46.53	0.68
T5-flan-base Triples-to-Reasoning	0.970	67.55	0.78
T5-flan-base Reasoning-to-Text	0.948	59.67	0.71

The evaluation results summarized in Table 1 indicate that the augmented dataset with an intermediate reasoning step yielded good performance in thist low-resource setting. While T5-small struggled to

[5]Downloaded from https://huggingface.co/datasets/webnlg-challenge/web_nlg.

generalize from the augmented data, T5-base demonstrated better performance. In fact, the pipeline model based on flan-T5-small did not match the end-to-end model, but the pipeline model based on flan-T5-base outperformed it. This could be attributed to the larger parameter count of T5-base.

Table 2 shows that both models perform similarly in the 'triples-to-reasoning' stage, but the 'reasoning-to-text' stage presents a bottleneck for both. However, the bottleneck is less pronounced in the case of flan-T5-base, which increases the performance ceiling, helping the pipeline model surpass the end-to-end model.

Confirming [2], we find that the flan-T5-base Single Model generates more hallucinated content compared to the pipelined model, which reduces hallucinations. For example, given the input triples:

```
<H>Death_on_a_Factory_Farm<R>director<T>61.0
<H>Death_on_a_Factory_Farm<R>broadcastedBy<T>61.0
<H>Death_on_a_Factory_Farm<R>editor<T>61.0
<H>Death_on_a_Factory_Farm<R>musicComposer<T>61.0
<H>Death_on_a_Factory_Farm<R>runtime<T>61.0
<H>Death_on_a_Factory_Farm<R>director<T>61.0
```

the single model generates:

```
The director of Death on a Factory Farm is Tom Simon. The director of Death
on a Factory Farm is Geof Bartz. The director of Death on a Factory Farm is
Tom Simon. The director of Death on a Factory Farm is Tom Simon. The director
of Death on a Factory Farm is Tom Simon. The director of Death on a Factory
Farm is Geof Bartz. The director of Death on a Factory Farm is Tom Simon.
The director of Death on a Factory Farm is Tom Simon. The director of Death
on a Factory Farm is Tom Simon.
```

while the pipelined model generates:

```
Tom Simon is the director of Death on a Factory Farm which was broadcasted
by HBO and edited by Geof Bartz. Jamie Lawrence is the music composer of
Death on a Factory Farm.
```

The pipelined model stays more accurate by strictly adhering to the input data and avoiding unrelated details and repetitions.

6 Conclusion

In this study, we addressed the challenges of graph-to-text generation in low-resource settings by introducing an intermediate reasoning step facilitated by GPT-3.5 Turbo. By dividing the task into graph-to-reasoning and reasoning-to-text subtasks and using knowledge distillation, we trained smaller FLAN-T5 models (small and base variants) to replicate reasoning capabilities. Our results demonstrate that the pipelined FLAN-T5-base model outperforms the single-model baseline, emphasizing the effectiveness of our approach, though FLAN-T5-small showed limited improvements due to model capacity constraints.

Our study is subject to a few limitations. Computational restrictions prevented statistical significance testing, exploration of larger models, and broader architectural variations beyond FLAN-T5. We also lacked ablation studies to isolate the contributions of individual components and did not explore the method's generalizability to other tasks. These constraints highlight areas for future research and underscore the need for further experimentation to validate and extend our findings.

Acknowledgments and Disclosure of Funding

This work was part of a course project at ETH Zurich, and we thank Mrinmaya Sachan, Assistant Professor, for his guidance, and Shehzaad Dhuliawala, PhD student and IBM Fellow, for his feedback. No funding or competing interests are declared.

References

[1] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1026. URL https://aclanthology.org/P18-1026.

[2] Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1052. URL https://aclanthology.org/D19-1052.

[3] Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1052. URL https://aclanthology.org/D19-1052.

[4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[5] Marco Damonte and Shay B. Cohen. Structural neural encoders for amr-to-text generation, 2019.

[6] Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023.

[7] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2, 2005. doi: 10.1109/IJCNN.2005.1555942.

[8] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023.

[9] Stephen Heller, Alan McNaught, Stephen Stein, Dmitrii Tchekhovskoi, and Igor Pletnev. Inchi-the worldwide chemical structure identifier standard. *Journal of cheminformatics*, 5:1–9, 2013.

[10] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[11] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.223. URL https://aclanthology.org/2021.findings-acl.223.

[12] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[13] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers, 2022.

[14] Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1014. URL https://aclanthology.org/P17-1014.

[15] Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Bin Sun, Xinglin Wang, Heda Wang, and Kan Li. Turning dust into gold: Distilling complex reasoning capabilities from llms by leveraging negative data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38:18591–18599, Mar. 2024. doi: 10.1609/aaai.v38i17.29821. URL https://ojs.aaai.org/index.php/AAAI/article/view/29821.

[16] Amit Moryossef, Yoav Goldberg, and Ido Dagan. Step-by-step: Separating planning from realization in neural data-to-text generation. *CoRR*, abs/1904.03396, 2019. URL http://arxiv.org/abs/1904.03396.

[17] Amit Moryossef, Yoav Goldberg, and Ido Dagan. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1236. URL https://aclanthology.org/N19-1236.

[18] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

[19] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. Knowledge graphs: Opportunities and challenges, 2023. URL https://arxiv.org/abs/2303.13948.

[20] Maja Popović. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4770. URL https://aclanthology.org/W17-4770.

[21] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

[22] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[23] Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. Enhancing AMR-to-text generation with dual graph representations. 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 3183–3194, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1314. URL https://aclanthology.org/D19-1314.

[24] Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. Modeling global and local node contexts for text generation from knowledge graphs, 2020.

[25] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation, 2021.

[26] Leonardo F. R. Ribeiro, Yue Zhang, and Iryna Gurevych. Structural adapters in pretrained language models for amr-to-text generation, 2021.

[27] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1150. URL https://aclanthology.org/P18-1150.

[28] Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1627–1637, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1151. URL https://aclanthology.org/P18-1151.

[29] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

[30] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[31] Shaopeng Wei, Yu Zhao, Xingyan Chen, Qing Li, Fuzhen Zhuang, Ji Liu, and Gang Kou. Graph learning and its applications: A holistic survey, 2023.

[32] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[33] Bohao Yang, Chen Tang, Kun Zhao, Chenghao Xiao, and Chenghua Lin. Effective distillation of table-based reasoning ability from llms. *arXiv preprint arXiv:2309.13182*, 2023.

[34] Shuzhou Yuan and Michael Färber. Evaluating generative models for graph-to-text generation, 2023. URL https://arxiv.org/abs/2307.14712.

[35] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019. URL http://arxiv.org/abs/1904.09675.

[36] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. Distilling mathematical reasoning capabilities into small language models. *Neural Networks*, 179:106594, 2024. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2024.106594. URL https://www.sciencedirect.com/science/article/pii/S0893608024005185.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims presented in the abstract and introduction accurately summarize the paper's contributions and the scope of our research.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion of the limitations can be found in the Conclusion (Section 6).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper focuses on an empirical approach to enhancing graph-to-text generation using smaller models and does not present theoretical results or proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides a thorough explanation of the training processes, including both task-adaptive pre-training and fine-tuning of the two models in the pipeline. These details can be found in Section 3. Additionally, we present the hyperparameters used for both processes in detail in Section 4. The prompt template we used for generating the synthetic reasoning dataset will be provided as part of the code, due to space limitations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code to reproduce the experiments can be found in our GitHub repository: https://github.com/davidguzmanp/Graph-to-Text-LLM-with-dataset-augmentation. The repository includes Jupyter notebooks that facilitate the reproduction of our experiments, including data pre-processing, loading the pre-trained models, and other essential steps. The repository also provides detailed instructions and commands for setting up the environment. We utilized the open-source WebNLG dataset, and a link to download it is included in the README file.

   Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The hyper-parameters are given in Section 4, and the data-splits are mentioned in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to computational limitations, we were unable to conduct multiple runs of the experiments to compute and report error bars or other measures of statistical significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: We conducted the experiments across multiple platforms that offer free GPU resources, which led to inconsistent runtimes and computational capabilities. Due to the variability in resource allocation and execution time on these platforms, we are unable to provide specific information on the computer resources required to reproduce the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Our research adheres to the NeurIPS Code of Ethics as it does not involve human subjects and relies solely on publicly available datasets, ensuring compliance with privacy and consent guidelines. Given the nature of our graph-to-text translation task, the concerns regarding bias and societal impact are not applicable, as the task is designed to minimize hallucinations.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our work focuses on a graph-to-text translation task using publicly available datasets, which minimizes the potential for societal impacts. As such, there are no significant positive or negative societal impacts to discuss regarding the research performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not involve the release of models or datasets that pose significant risks for misuse. The data utilized in our research is sourced from the open WebNLG dataset, which is publicly available and does not require specific safeguards for responsible release.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We do not use code from another paper; however, we adapt the methodology from Ribeiro et al. [25] for our dataset as a baseline. We mention that in Section 3 of our paper. For the WebNLG dataset, which is openly available under Creative Commons Attribution Share Alike 3.0 license. We comply with its licensing terms and provide the appropriate citation and link to the source in Section 4.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The paper does not introduce new assets. We do not publish the synthetic dataset due to OpenAI's terms of service and usage policies.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The study did not involve direct interaction with human participants or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The study did not involve direct interaction with human participants or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.