Iterative Graph-to-Text Generation with Contextualization for Scientific Abstracts

Haotong Wang, Liyan Wang, Yves Lepage
大学院情報生産システム研究科, 早稲田大学
Graduate School of Information, Production and Systems, Waseda University
{wanghaotong0925, wangliyan0905}@toki.waseda.jp yves.lepage@waseda.jp

Abstract

We propose an iterative graph-to-text generation method to produce coherent scientific abstracts from paragraph-level knowledge graphs. The method segments the graphs into smaller, context-specific components using functional labels, which guide each generation step and influence subsequent outputs. Experimental results demonstrate that fine-tuning the proposed method enhances the alignment of Large Language Models (LLMs) with target semantics. Moreover, incorporating functional labels and iterative generation further improves semantic accuracy, structural clarity, and logical organization, providing a scalable solution for high-quality abstract generation.

1 Introduction

Knowledge graphs facilitate the construction of rich semantic information by systematically organizing and interlinking multidimensional entities and relationships within a network-based structure [1]. Typically composed of triples (head entity, relation, tail entity), they provide a more direct representation of the core content in scientific abstracts, where the generation of abstract texts based on knowledge graphs has emerged as a prominent research focus, with applications in assisting academic writing [2] and enhancing the understanding of scientific research [3].

Current research on graph-to-text generation primarily revolves around Graph Neural Networks (GNNs) and Large Language Models (LLMs). GNN-based approaches [2, 4] focus on the structured representation of content, constructing more precise relational networks among entities. In contrast, LLM-based methods [5, 6] leverage the powerful generative capabilities of large language models to combine and integrate triple-based content, producing more coherent and fluent paragraph-level text. In scientific abstracts, semantically complex paragraphs often lead to knowledge graphs with intricate structures. Challenges such as high computational costs and content forgetting may arise when using traditional GNNs to generate text from such graphs [7]. On the other hand, LLMs, despite their strong generative capabilities, lack structural awareness of the graph, making it crucial to ensure content consistency and paragraph coherence during generation. Once linearized, this issue stems from triples in a knowledge graph, becoming discrete and lacking a strong logical flow. While LLMs can effectively organize language at the sentence level, an excessive number of triples can hinder the overall quality of the generated text.

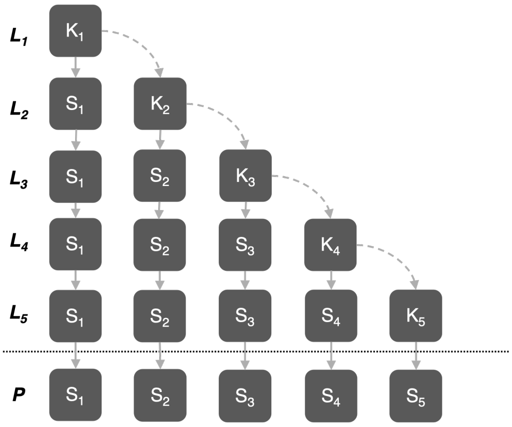


Figure 1 Workflow of the Iterative Graph-to-Text Generation. L_i: Paragraph segmentation function label, K_i: Sentence knowledge graph corresponding to the label, S_i: Generated sentence, P: Entire paragraph.

To address the above issues, we propose an iterative graph-to-text generation method with contextualization for scientific abstracts. As shown in Figure 1, first, we divide the abstract into functional segments and apply strict structural control using labels (label hard control) [8]. Then, we generate sentences step by step based on the given labels and knowledge graphs. During this process, the input at each step combines the previously generated sentence with the following knowledge graph, following the paragraph's structural order. This method ensures the coherence and consistency of the paragraph by integrating labels with knowledge graphs. In addition, the generation process uses the previous sentence as context to guide the expression of the knowledge graph content, prevent semantic deviations, and eliminate ambiguities.

2 Methodology

2.1 Iterative Graph-to-Text Generation

The knowledge graph consists of commonly defined triples $t = (h, r, t)$, where h is the head entity, r is the relation, and t is the tail entity. These triples for a scientific abstract are segmented using L_i (labels), which categorize them based on paragraph functions (e.g., background, objective, methods). The segmented triples form a set $\mathcal{K} = (K_1, K_2, \ldots, K_i)$, where K_i represents the triples under a specific label. The objective of this paper is to generate a set of sentences $\mathcal{S} = (S_1, S_2, \ldots, S_i)$ from $\mathcal{K}$. Finally, these generated sentences $\mathcal{S}$ are sequentially combined to construct the scientific abstract paragraph P.

A scientific abstract distills the essence of the research presented in an article, providing a concise summary of the study's objectives, methodology, results, and significance. We divide the abstract into five labels: Background, Objective, Methods, Results, and Conclusions [8]. These labels can segment long paragraphs into smaller units, which helps reduce the complexity of text generation and eliminate ambiguities. They are also used in the iterative generation method proposed in this paper, serving to expand the dataset during training.

As described in Algorithm 1, if the starting label is L_i, we first generate descriptive text for the label L_i using its corresponding knowledge graph K_i. Generating short text for each label proves more reliable than generating the entire paragraph from a single knowledge graph. Next, we concatenate the sentence S_i generated for the label L_i with the knowledge graph K_{i+1} of the subsequent label. The previously generated text serves as contextual guidance, helping to correct the understanding of the content. This process continues iteratively until the text under the current label is generated, repeating the process until the entire paragraph P is complete.

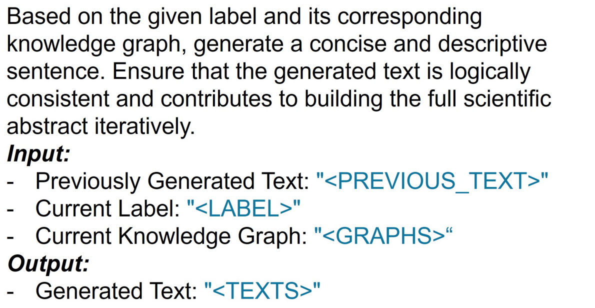
Algorithm 1 Iterative Text Generation from Knowledge Graphs

```
Input: Knowledge Graphs $K_1, K_2, \ldots, K_n$ (ordered, labeled)
Output: Generated Paragraph $P$
Initialize $P \leftarrow \emptyset$        ▷ Start with an empty paragraph
for each label $L_i$ in $K$ do
    Generate sentence $S_i$ for label $L_i$ using knowledge graph $K_i$
    Concatenate $P \leftarrow P \| S_i$
    if more labels remain then
        Update $K_{i+1} \leftarrow P \cup K_{i+1}$
    end if
end for
return $P$
```

2.2 Prompt Design

We fine-tune the FLAN-T5 model [9] for the specific task in this work. As a sequence-to-sequence model, FLAN-T5 utilizes instruction tuning to enable efficient and accurate text generation, achieving precise mapping between inputs and outputs.

As shown in Figure 2, we design prompts based on the task requirements. The input contains three key pieces of information: <PREVIOUS_TEXT>, <LABEL>, and <GRAPHS>. To better distinguish the content, the label is marked with special tokens <l> and </l>, while the triples in the knowledge graph are marked with <h>, <r>, and <t> to represent the head, relation, and tail, respectively. For iterative generation, the content of these three key components is continuously updated until the entire paragraph is generated.



Based on the given label and its corresponding knowledge graph, generate a concise and descriptive sentence. Ensure that the generated text is logically consistent and contributes to building the full scientific abstract iteratively.

Input:

- Previously Generated Text: "<PREVIOUS_TEXT>"
- Current Label: "<LABEL>"
- Current Knowledge Graph: "<GRAPHS>“

Output:

- Generated Text: "<TEXTS>"

Figure 2 Prompt design for iterative graph-to-text generation.

3 Dataset

We utilize the ACL Abstract Graph Dataset (ACL-AGD) in this work [1]. The ACL-AGD comprises 35,063 abstracts collected from the ACL Anthology's BibTeX database. It features various research works in computational linguistics and natural language processing, ranging from conference proceedings and journal publications to selected papers from non-ACL events. The dataset spans nearly six decades of scholarly contributions, covering 1965 to 2023. Based on the functional segmentation of scientific abstracts [8], each triple is assigned a label, forming a quadruple.

4 Experiments

4.1 Evaluation Metrics

Building on prior work [5, 10], we evaluate our models using four widely adopted metrics: BLEU-4 [11], METEOR [12], ChrF++ [13], and ROUGE-L [14]. These metrics comprehensively assess the generated text, capturing its linguistic accuracy and semantic relevance compared to the corresponding target texts.

4.2 Generation Evaluation

To evaluate the generation performance on the ACL-AGD dataset, we compared several methods, including GPT-3 and ChatGPT, as reported in [10]. As shown in Table 1, These two models operate in a zero-shot learning paradigm, which means they lack the contextual understanding provided by specific annotations such as paragraph labels. Consequently, they struggle to accurately map the knowledge graphs to the target text, leading to deviations in generation quality. This limitation is reflected in their lower evaluation metrics scores compared to other methods.

We further analyzed the impact of incorporating paragraph labels in non-iterative and iterative settings. The results demonstrate a significant improvement in generation quality when labels are included, as evident from the increase in all evaluation metrics. For instance, in the non-iterative setting, adding labels improved BLEU from 8.54 to 11.75 and METEOR from 29.72 to 34.64. This highlights the effectiveness of labels in providing structural and semantic guidance for more accurate text generation.

Table 1 Comparison of generation performance on ACL-AGD.

Methods	BLEU	METEOR	CharF++	ROUGE
GPT-3 [10]	$7.52_{\pm1.81}$	$30.16_{\pm1.73}$	$38.61_{\pm1.89}$	$35.45_{\pm1.92}$
ChatGPT [10]	$10.94_{\pm2.11}$	$32.23_{\pm1.84}$	$44.89_{\pm1.96}$	$37.67_{\pm2.07}$
Non-iterative				
- w/o label	$8.54_{\pm1.93}$	$29.72_{\pm1.79}$	$36.66_{\pm1.84}$	$32.46_{\pm2.09}$
- with label	$11.75_{\pm1.94}$	$34.64_{\pm1.72}$	$42.16_{\pm2.03}$	$37.31_{\pm1.85}$
Iterative				
- w/o label	$9.40_{\pm2.02}$	$31.97_{\pm1.85}$	$40.43_{\pm1.91}$	$36.34_{\pm1.96}$
- with label†	$12.74_{\pm1.78}$	$35.29_{\pm1.88}$	$44.25_{\pm1.95}$	$38.90_{\pm1.89}$

† denotes our proposed method.

Finally, we examined the difference between iterative and non-iterative approaches. Iterative methods consistently outperformed their non-iterative counterparts, showcasing their ability to refine generated text progressively. Among all approaches, the combination of the iterative method and paragraph labels achieved the best performance, with a BLEU score of 12.74 and a METEOR score of 35.29. This demonstrates the superiority of our proposed iterative approach with labels in aligning the generated text more closely with the target abstract. Appendix A provides an example of iterative generation.

4.3 Impact of Generation Length

The results in Figure 3 highlight a key aspect of generation performance related to output length (the number of sentences). GPT-3 and ChatGPT exhibit longer and more variable outputs, likely contributing to their suboptimal generation quality. These models tend to fill the text with additional content that deviates semantically from the target, indicating a lack of precise alignment with the knowledge graph.

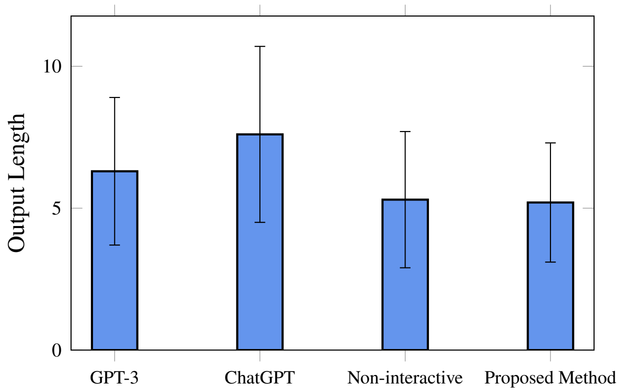


Figure 3 Comparison of Output Length.

In contrast, our fine-tuning-based approach, both with

[1] http://lepage-lab.ips.waseda.ac.jp/projects/scientific-writing-aid/

and without iteration (with label), demonstrates a more stable and concise output length. This consistency enables the generated text to remain focused on the core content of the knowledge graph, ensuring better alignment with the target text. The reduced variability in output length underscores the effectiveness of our method in maintaining semantic relevance and structural precision.

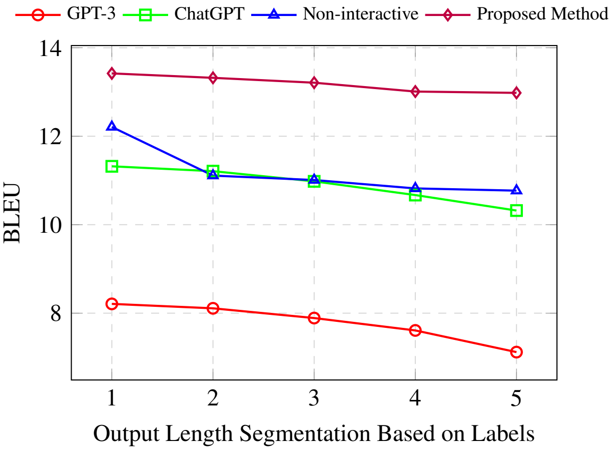


Figure 4 Variation of BLEU with Increasing Output Length.

As shown in Figure 4, the BLEU scores of all methods decrease as the generated text length increases, reflecting the challenge of maintaining semantic alignment with more extended outputs. GPT-3 and ChatGPT exhibit a particularly sharp decline in BLEU scores, suggesting that these models struggle to handle extended sequences without significant semantic drift. This behavior highlights their limitations in preserving coherence over more extended text generations.

Interestingly, non-iterative methods show a moderate decline but cannot adapt contextually to the increasing complexity of longer sequences. In comparison, our proposed iterative method demonstrates superior stability. While there is a gradual decrease in BLEU scores, the decline is significantly less steep. This indicates that the iterative approach effectively integrates contextual information, allowing it to generate text that remains closely aligned with the semantic and structural requirements of the target text.

5 Conclusion

In this paper, we proposed an iterative graph-to-text generation approach tailored for generating scientific abstracts. Our method uses functional labels to leverage contextual information by segmenting paragraph-level knowledge graphs into smaller components. These labels guide the generation process iteratively, where the text generated from one label and its associated knowledge graph informs the subsequent label and graph, continuing until the entire paragraph is generated. This iterative design allows the model to maintain coherence and align closely with the structure and semantics of the target abstract.

Our experiments reveal several key findings. First, LLMs, such as GPT-3 and ChatGPT, struggle to fully comprehend and effectively utilize knowledge graphs in zero-shot settings, resulting in significant semantic drift and structural inconsistencies. This underscores the importance of fine-tuning, which enables LLMs to interpret graph-encoded information better and align generated text with the target output. Second, including paragraph labels and iterative generation significantly improves semantic accuracy and structural clarity. Labels provide crucial guidance for mapping the graph to the text, while the iterative approach corrects semantic errors progressively, ensuring a coherent flow and logical organization throughout the paragraph.

The proposed method offers a scalable framework for generating high-quality scientific abstracts by ensuring semantic fidelity and paragraph-level structural clarity, effectively addressing key challenges in graph-to-text generation. It paves the way for further advancements in context-aware generation methods, enhancing the quality of machine-generated content in knowledge-intensive domains.

References

[1] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. **IEEE Transactions on Neural Networks and Learning Systems**, Vol. 33, No. 2, pp. 494–514, 2022.

[2] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text Generation from Knowledge Graphs with Graph Transformers. 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)**, pp. 2284–2293, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] Sonal Gupta and Christopher Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In Haifeng Wang and David Yarowsky, editors, **Proceedings of 5th International Joint Con-**

ference on Natural Language Processing, pp. 1–9, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[4] Anthony Colas, Mehrdad Alvandipour, and Daisy Zhe Wang. GAP: A graph-aware language model framework for knowledge graph-to-text generation. In **Proceedings of the 29th International Conference on Computational Linguistics**, pp. 5755–5769, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[5] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. In **Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI**, pp. 211–227, Online, November 2021. Association for Computational Linguistics.

[6] Zixiaofan Yang, Arash Einolghozati, Hakan Inan, Keith Diedrick, Angela Fan, Pinar Donmez, and Sonal Gupta. Improving text-to-text pre-trained models for the graph-to-text task. In **Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)**, pp. 107–116, Dublin, Ireland (Virtual), 12 2020. Association for Computational Linguistics.

[7] Fan Zhou and Chengtai Cao. Overcoming catastrophic forgetting in graph neural networks with experience replay. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 35, No. 5, pp. 4714–4722, May 2021.

[8] Sérgio Gonçalves, Paulo Cortez, and Sérgio Moro. A deep learning classifier for sentence classification in biomedical and computer science abstracts. **Neural Computing and Applications**, Vol. 32, No. 11, pp. 6793–6807, 2020.

[9] 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. **Journal of Machine Learning Research**, Vol. 25, No. 70, pp. 1–53, 2024.

[10] Shuzhou Yuan and Michael Faerber. Evaluating generative models for graph-to-text generation. In Ruslan Mitkov and Galia Angelova, editors, **Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing**, pp. 1256–1264, Varna, Bulgaria, September 2023. INCOMA Ltd., Shoumen, Bulgaria.

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[12] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, **Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization**, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[13] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, **Proceedings of the Tenth Workshop on Statistical Machine Translation**, pp. 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[14] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Text Summarization Branches Out**, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

A Iterative Generation Process

Table 2 illustrates the iterative generation process, where sentences S_i are generated sequentially based on knowledge graphs K_i. Initially, S_1 is generated based on K_1, representing the background information. Subsequently, S_1 is combined with K_2 to produce S_2, corresponding to the objective. This process continues by iteratively combining the previous sentence S_{i-1} with a new knowledge graph K_i, thereby completing the iterative process.

Table 2 Iterative Generation Process

K_1: <l>BACKGROUND</l><h>digital language archiving services <r>HYPONYM-OF <t>disciplines <h>low-resourced and endangered languages <r>USED-FOR <t>speech and language data
S_1: Digital language archiving services are essential components of various linguistic disciplines, particularly for managing speech and language data of low-resourced and endangered languages.
S_1 + K_2: <l>OBJECTIVE </l><h>It <r>USED-FOR <t>common digital language archive infrastructures
S_2: It is aimed at enhancing common digital language archive infrastructures to improve data availability and support broader research needs.
S_2 + K_3: <l>METHODS</l><h>digital language archive services <r>CONJUNCTION <t>language data <h>It <r>USED-FOR <t>language models <h>language models <r>USED-FOR <t>speech and natural language processing <h>GORILLA <r>USED-FOR <t>Transcription Bottleneck <h>GORILLA <r>USED-FOR <t>general Language Resource Bottleneck <h>Transcription Bottleneck <r>PART-OF <t>language documentation <h>general Language Resource Bottleneck <r>PART-OF <t>speech and language technologies
S_3: Digital language archive services are combined with language data and utilized to support language models for speech and natural language processing tasks. This paper propose GORILLA that effectively mitigates both the transcription bottleneck in language documentation and the general language resource bottleneck in speech and language technologies.