International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Comprehensive Study of Graph-to-Text Generation

K. Vyshnavi Sri Sindhu

Computer Science and Engineering Department,
GMR Institute of Technology, Rajam,
Andhra Pradesh, India

ABSTRACT –

Graph-to-text generation in NLP involves converting structured data from graphs into coherent human-readable text. The process integrates graph representation learning, attention mechanisms, and text generation models, often leveraging transformers. This burgeoning field facilitates communication across applications like data visualization and knowledge graph summaries. Advancements in graph representation learning and attention mechanisms promise enhanced accuracy and fluency in generated text, bridging the gap between structured data and human interpretation. This area is pivotal for effective communication, translating diverse graphs into understandable narratives for applications in data analysis and knowledge representation.

Keywords - Graph-to text generation; Attention mechanism; Natural Language Processing (NLP); transformer model.

1. Introduction

Graph-to-text generation, a dynamic domain within natural language processing (NLP), involves translating structured data, often in graph form, into coherent human-readable text. This process integrates graph representation learning, attention mechanisms, and text generation models, acting as a crucial bridge between complex structured data and clear human interpretation across diverse applications. Graph representation learning forms the foundation, extracting meaningful features from diverse graph components to capture both structure and semantics. This step enables the model to understand relationships and patterns within the graph, forming a knowledge base for effective translation into human-readable text.

Attention mechanisms play a pivotal role, guiding the model to focus on relevant graph elements during text generation. Similar to human communication, attention mechanisms prioritize key information, ensuring contextually appropriate and coherent output. They serve as a guiding force in steering the model towards meaningful narratives. Text generation models, often based on transformers, leverage learned graph features to craft contextually appropriate and coherent text. Combining insights from graph representation learning with the focus provided by attention mechanisms, these models navigate the interplay between structure and semantics, producing human-readable sentences that encapsulate the essence of structured data.

Graph-to-text generation addresses critical needs in data analysis and knowledge representation. As structured data grows in volume and complexity, translating it into accessible narratives becomes essential for informed decision-making and effective communication. The applications are diverse, including data visualization descriptions and knowledge graph summarization. In data visualization, the technology automates the generation of explanations and insights from complex visual representations, empowering non-experts to derive meaningful information. In knowledge graph summarization, it provides a concise means of conveying rich knowledge to a broader audience, with transformative effects in fields like healthcare, finance, and research.

Despite its promise, challenges persist. Balancing faithful representation of the underlying graph with coherent and digestible text remains a research challenge. Adapting models to handle diverse graph types, from semantic networks to hierarchical structures, requires flexibility. Scalability to handle large-scale graphs is also a hurdle actively addressed by researchers. Graph-to-text generation stands at the intersection of structured data and human communication, revolutionizing data analysis, knowledge representation, and communication across domains. With foundational pillars of graph representation learning, attention mechanisms, and text generation models, ongoing advancements hold promise for improving accuracy and fluency. In essence, this field is vital, dynamic, and holds remarkable potential for the future of data-driven communication.

2. Literature Survey

Koncel-Kedziorski (2019) main purpose is to generating coherent multi-sentence texts from knowledge graphs. They proposed a novel graph transforming encoder that capitalizes on knowledge graph structure without imposing constraints. Their end-to-end system enhances scientific text generation, surpassing other methods. Additionally, they explore using information extraction systems to provide context through knowledge graphs in attention-based encoders. It demonstrates improved text quality and document structure, valuable for scientific domains, as evidenced by both automated and human evaluations. [1]. Kalyan (2021) presented a comprehensive survey of Transformer-based pretrained language models (T-PTLMs) in the realm of NLP. Their aims include providing insights into self-supervised learning, pretraining methods, embeddings, and downstream adaptation. They introduced a novel T-PTLM taxonomy and offer valuable resources for working with these models. Future research directions are also highlighted, making this survey a vital reference for staying current in T-PTLMs, NLP [2]. Li,L.,(2022) introduced Structure-Aware Cross-Attention (SACA) and Dynamic Graph Pruning (DGP) mechanisms for graph-to-text generation,. They improved performance on datasets like LDC2020T02 and ENT-DESC with minimal computational cost. This work enhances natural language descriptions from structured data like Knowledge Graphs, contributing to graph interpretability and knowledge-based question answering within the encoder-decoder framework [3]. Ribeiro (2020) presented neural models for text generation from knowledge graphs, integrating global and local node contexts through cascaded and parallel node aggregations. Outperforming state-of-the-art models on two datasets, it employs varying layers and attention heads in graph encoders, demonstrating effectiveness. [4]. Ke,p.,(2021) aimed to overcome limitations in knowledge graph-to-text generation by introducing JointGT, a novel graph-text joint representation learning model. Their objectives involve preserving graph structure, enhancing alignment between graph and text, and introducing new pre-training tasks. JointGT achieves state-of-the-art performance on knowledge graph-to-text datasets**.**[5]. Zhao(2020) introduced generating natural language descriptions from graph-structured data by encoding and decoding. It presents DUA-LENC, a dual encoding model that considers both the graph structure and the linear text structure to enhance text quality. Additionally, a neural planner is introduced to create a content plan from the graph, reducing the structural gap and combining the strengths of graph and sequential encoders [6].Cai(2020) presented the Graph Transformer model for enhanced graph-to-sequence learning, overcoming limitations in existing graph neural networks by enabling direct communication between distant nodes. Superior performance is demonstrated in text generation tasks, including Abstract Meaning Representation and syntax-based neural machine translation. Internal analysis highlights the multi-head attention mechanism's contribution to improvement gains. [7].Guo,Z.,(2019) introduced Densely Connected Graph Convolutional Networks (DCGCN) for improved graph-to-sequence learning, outperforming state-of-the-art models in AMR-to-text generation and syntax-based neural machine translation. [8]. Guo,Q.,(2020) proposed the P2 approach, integrating R-GCN planning with pre-trained Seq2Seq model (T5) for knowledge graph-to-text generation. Achieving 1st place in WebNLG+ 2020 Challenge underscores its effectiveness. [9]. Lin,Y.,(2023) P2 approach combines R-GCN planning and a pre-trained Seq2Seq model for knowledge graph-to-text generation, securing 1st place in the WebNLG+ 2020 Challenge for English RDF-to-text. [10]. Colas(2022) presented a graph-aware language model for knowledge graph-to-text generation, minimizing reliance on pre-training. Their framework incorporates a mask structure for neighborhood information and a type encoder adjusting graph-attention weights. Interchangeable components yield interpretable models, exhibiting competitiveness on benchmarks with fewer parameters and no additional pre-training tasks. [11]. Yang(2022) introduced ADGCN for generating keywords from short texts via graph-to-sequence learning, addressing challenges such as topic dependence and text structure issues. Evaluated on the KP20k dataset, it demonstrates effectiveness in generating topic keywords and mitigating data disturbances. [12].Wang(2023) enhanced graph-to-text generation with pretrained language models (PLMs) by introducing the Relational Orientation Attention (ROA) module and utilizing (knowledge subgraph, text) pairs for PLM pretraining. These strategies address triplet structure information loss, enhance PLMs' structured data handling, and improve few-shot learning, as validated through experiments on diverse datasets. [13].Yun(2022) introduced Graph Transformer Networks (GTNs) and Fast Graph Transformer Networks (FastGTNs) to overcome limitations of existing Graph Neural Networks (GNNs) on mis specified or heterogeneous graphs. GTNs identify meta-paths and multi-hop connections, and FastGTNs enhance scalability and speed, achieving state-of-the-art results in node classification across diverse graph types. [14].Brauwers(2021) provided an extensive overview of attention mechanisms in deep learning models, encompassing diverse domains and tasks. It offers a structured framework with a common notation and taxonomy for various attention mechanisms. Additionally, the paper delves into evaluation measures and strategies for characterizing attention model structures, contributing to a comprehensive understanding of attention mechanisms in deep learning.[15].

3. Methodology

It mainly focuses on creating meaningful text from knowledge extracted by information extraction (IE) systems. It formulates the challenge of abstract generation by using a knowledge graph that considers both global and local characteristics. The AGENDA dataset includes 40,000 paper titles and abstracts, where the SciIE system is employed for entity recognition and relation annotations. This dataset is divided into 38,720 training, 1000 validation, and 1000 test datapoints. The approach utilizes graphs to represent scientific articles, emphasizing the importance of capturing information and connections through this structured representation.

Preprocessing

Information extraction is conducted from both the knowledge graph and input text. The knowledge graph is structured as a collection of vertices and edges, with each vertex corresponding to entities or relations identified through SciIE annotations. Entities within the knowledge graph often involve multi-word expressions, and their embeddings are derived by employing a bidirectional RNN across the word embeddings of each constituent word in the entity phrase. The resulting encodings comprehensively represent entities, relations, and the overarching node within the graph, contextualized by their intricate relationships. Concurrently, the input text undergoes thorough processing to extract pertinent information, culminating in the generation of a document plan. This plan functions as a meticulously structured representation of the content, ensuring the assimilation of crucial details. It serves as a foundation for the subsequent coherent generation of text, seamlessly integrating knowledge gleaned from both the knowledge graph and the processed input text.

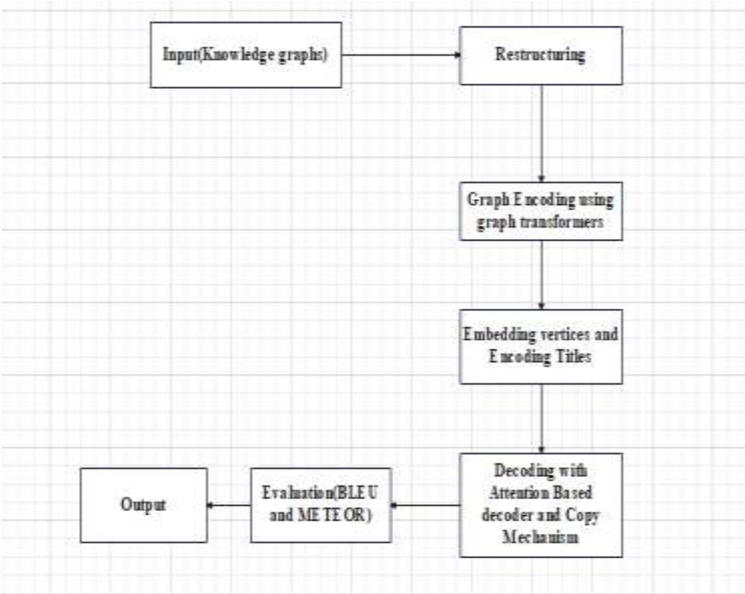

Fig. Structure

Architecture

The approach centers on refining knowledge graph structures and introducing the Graphwriter model for text generation, adept at handling input titles and associated knowledge graphs. Initially, the AGENDA dataset's knowledge graphs undergo restructuring, preserving labels by substituting edges with forward and reverse direction vertices. This maintains edge directionality in connections to entity vertices.The Graphwriter architecture adopts an encoder-decoder model, processing both the title and knowledge graph. The encoding employs a bidirectional recurrent neural network (RNN) and the Graph Transformer architecture, amalgamating self-attention and a transformer-style framework. N-headed self-attention contextualizes vertex representations, supplemented by block networks for enhanced information propagation.The encoding phase unfolds in three steps: utilizing graph transformers to encode titles and graphs, embedding vertices, and encoding titles. The Graph Transformer model employs self-attention and block networks to capture entities, relations, and global context within the graph structure. Embeddings for entities, relations, and the global node are created, with bidirectional RNNs processing multi-word entity phrases and titles.Decoding incorporates an attention-based decoder with a copy mechanism, factoring in both the knowledge graph and title. Context vectors, derived from multi-headed attention with the decoder's hidden state, enable the model to copy from input or select vocabulary for text generation. This meticulous methodology seamlessly integrates knowledge and title information, showcasing a holistic comprehension of the underlying content.

Evaluation metrics

Evaluation metrics such as BLEU (n-gram overlap measure) and METEOR (machine translation with paraphrase and language-specific considerations) were employed to evaluate the system's performance.

BLEU (Bilingual Evaluation Understudy) is an automatic evaluation metric commonly used in machine translation tasks to measure the quality of generated translations.

$$\mathrm{BLEU}=\mathrm{BP}\times\exp\left(\sum_{n=1}^{N} w_n\cdot\log(p_n)\right)$$

where:

BP (Brevity Penalty): Penalty term to address the issue of short translations.

N: Maximum n-gram order considered (typically 4).

Wn: Weight assigned to the precision at n-grams.

Pn: Modified precision at n-grams, calculated as the ratio of the number of matching n-grams in the candidate translation to the total number of n-grams in the candidate translation.

The **METEOR** (Metric for Evaluation of Translation with Explicit ORdering) metric is another evaluation measure used to assess the quality of machine-generated translations. The formula for calculating METEOR involves precision, recall, and a penalty term for unigram matching.

$$\mathrm{METEOR}=(1-\beta)\cdot\mathrm{precision}+\beta\cdot\mathrm{recall}-\gamma\cdot\mathrm{Penalty}$$

Where:

β: parameter that controls the importance of precision and recall. It is typically set to 1.0.

γ: parameter that controls the penalty for unigram matching. It is typically set to 0.5.

Penalty_count is the count of unigrams in the candidate that have exact matches in the reference.

4. Results and Discussions

The data evaluates natural language processing models using BLEU and METEOR scores. GraphWriter, employing Graph Transformers, Self-Attention Mechanism, and Bi-directional RNN, achieves BLEU scores of 14.3 and 18.8. Graph Transformers, featuring Self-Attention Mechanism, outperform other models with impressive BLEU scores of 29.8 and 35.1. Transformers with Self-Attention also demonstrate strong performance, scoring 27.85 and 38.95. METEOR scores provide additional insight into the models' language generation proficiency, collectively illustrating their effectiveness across diverse algorithms and attention mechanisms in natural language processing tasks.

TABLE: Comparison of different Graph-to-Text Generation models performed on AGENDA dataset.

Model	**BLEU**	**METEOR**
Graph Writer	14.3	18.8
GPT3	10.57	17.02
SMA-SCE	15.51	19.88
BART	23.65	25.19
Graph Transformers	29.8	35.1

5. Conclusion

In conclusion, this study proposes that Graph-to-text generation is a transformative field in natural language processing, translating structured graph data into coherent narratives. Integration of graph representation learning, attention mechanisms, and text generation models enables effective communication across applications. Representation learning extracts meaningful features, capturing structure and semantics, while attention mechanisms focus on relevant graph elements during text generation. This versatile technology applies to data visualization and knowledge graph summaries, promising improved accuracy and fluency. Despite challenges, graph-to-text generation emerges as a crucial bridge between diverse data representations, facilitating accessible information communication in data analysis and knowledge representation.

6. References

[1] Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., & Hajishirzi, H. (2019). Text generation from knowledge graphs with graph transformers. arXiv preprint arXiv:1904.02342.

[2] Tang, C., Guerin, F., Li, Y., & Lin, C. (2022). Recent advances in neural text generation: A task-agnostic survey. *arXiv preprint arXiv:2203.03047*.

[3] Kalyan, K. S., Rajasekharan, A., & Sangeetha, S. (2021). Ammus: A survey of transformer-based pretrained models in natural language processing. arXiv preprint arXiv:2108.05542.

[4] Li, L., Geng, R., Li, B., Ma, C., Yue, Y., Li, B., & Li, Y. (2022). Graph-to-text generation with dynamic structure pruning. *arXiv preprint arXiv:2209.07258*

[5] Ribeiro, L. F., Zhang, Y., Gardent, C., & Gurevych, I. (2020). Modeling global and local node contexts for text generation from knowledge graphs. *Transactions of the Association for Computational Linguistics*, *8*, 589-604.

[6] Ke, P., Ji, H., Ran, Y., Cui, X., Wang, L., Song, L., ... & Huang, M. (2021). Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. *arXiv preprint arXiv:2106.10502*.

[7] Zhao, C., Walker, M., & Chaturvedi, S. (2020, July). Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of theAssociationforComputationalLinguistics* (pp.481-2491).

[8] Cai, D., & Lam, W. (2020, April). Graph transformer for graph-to-sequence learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 05, pp. 7464-7471).

[9] Guo, Z., Zhang, Y., Teng, Z., & Lu, W. (2019). Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, *7*, 297-312.

[10] Guo, Q., Jin, Z., Dai, N., Qiu, X., Xue, X., Wipf, D., & Zhang, Z. (2020). 𝒫2: A Plan-and-Pretrain Approach for Knowledge Graph-to-Text Generation: A Plan-and-Pretrain Approach for Knowledge Graph-to-Text Generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)* (pp. 100-106).

[11] Lin, Y., Ruan, T., Liu, J., & Wang, H. (2023). A Survey on Neural Data-to-Text Generation. *IEEE Transactionson Knowledge and Data Engineering*.

[12] Colas, A., Alvandipour, M., & Wang, D. Z. (2022). GAP: A graph-aware language framework forknowledgegraph-to-textgeneration. *arXiv preprint arXiv:2204.06674*.

[13] Yang, Y., Sun, Y., Yang, D., Mei, G., & Wang, Z. (2022). Empower Keywords Generation for Short Texts with Graph-to-Sequence Learning. *Procedia Computer Science*, *207*, 100-109**.**

[14] Wang, T., Shen, B., Zhang, J., & Zhong, Y. (2023). Improving PLMs for Graph-to-Text Generation by Relational Orientation Attention. *Neural Processing Letters*, 1-17.

[15]. Yun, S., Jeong, M., Yoo, S., Lee, S., Sean, S. Y., Kim, R., ... & Kim, H. J. (2022). Graph Transformer Networks: Learning meta-path graphs to improve GNNs. *Neural Networks*, *153*, 104-119**.**

[16] Yuan, S., & Färber, M. (2023). Evaluating Generative Models for Graph-to-Text Generation. *arXiv preprint arXiv:2307.14712*.

[17] Ribeiro, L. F., Schmitt, M., Schütze, H., & Gurevych, I. (2020). Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*.

[18] Liu, Y., Zhang, L., Zhang, X., Jiang, Y., Zhang, Y., & Tu, K. (2021, August). Generalized supervised attention for text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 4991-5003).