GraphLSS: Integrating Lexical, Structural, and Semantic Features for Long Document Extractive Summarization

Anonymous ACL submission

Abstract

Heterogeneous graph neural networks have recently gained attention for long document summarization, modeling the extraction as a node classification task. Although effective, these models often require external tools or additional machine learning models to define graph components, producing highly complex and less intuitive structures. We present GraphLSS, a heterogeneous graph for long document extractive summarization, incorporating Lexical, Structural, and Semantic features. It defines two levels of information (words and sentences) and four types of edges (sentence semantic similarity, sentence occurrence order, word in sentence, and word semantic similarity) without requiring auxiliary learning models. Experiments on two benchmark datasets show that GraphLSS is competitive with top-performing graph-based methods, outperforming recent non-graph models. We release our code on <anonymized>.

1 Introduction

Extractive document summarization condenses documents into concise summaries by selecting only the most relevant sentences with key information to retain. One intuitive way for doing so is to model cross-sentence relations by using graphs. While some methods considered homogeneous graphs (Tixier et al., 2017; Xu et al., 2020), heterogeneous graph constructions have recently gained attention, showing high effectiveness on the task (Wang et al., 2020; Jia et al., 2020). Such graphs define more complex relationships between multiple semantic units and capture long-distance dependencies. Despite these graph structures have proven successful for long documents like scientific papers, many efforts have been made to propose more effective graph constructions. These methods differ in their definition of nodes, often requiring external tools or additional machine learning models (Cui et al., 2020), and in their definitions of edges, which despite being effective, may produce highly complex structures that reduce the intuitiveness of the resulting graphs (Zhang et al., 2022).

This paper introduces GraphLSS, a graph construction that avoids the need for external learning models to define nodes or edges. GraphLSS utilizes **L**exical, **S**tructural, and **S**emantic features, incorporating two types of nodes (sentences and words) and four types of edges (sentence order, sentences semantic similarity, words semantic similarity, and word–sentence associations). We limit word nodes to nouns and verbs for their high semantic richness. Our document graphs are processed with GAT (Veličković et al., 2018) models on two summary benchmarks, PubMed and arXiv, which are preprocessed and labeled by us.

Our contributions are: **i.** A new effective heterogeneous graph construction incorporating lexical, structural, and semantic features, **ii.** State-of-the-art results on both summary benchmarks compared to previous graph strategies and recent non-graph methods, **iii.** The preprocessed and labeled datasets, including the graph construction method, are shared on <anonymized> for reproducibility and collaboration.

2 Previous Work

Graph Structure Developing an effective graph structure for summarization has been challenging, leading to a proliferation of diverse approaches. Wang et al. (2020) proposed using word nodes to connect sentence nodes, with each word defining undirected associations with the sentences containing it. In turn, Jia et al. (2020) extended this by introducing named entity nodes and three other types of edges: directed edges for tracking the next named entity and word mentioned in a sentence, directed edges for entities and words occurring in a sentence, and undirected edges for sentence pairs with trigram overlap.

Topic-GraphSum (Cui et al., 2020) was one of the first attempts to apply graph strategies to long document extractive summarization. It integrated a joint neural topic model to discover latent topics in a document, defining these as intermediate nodes to capture inter-sentence relationships across various genres and lengths. SSN (Cui and Hu, 2021) defined a sliding selector network with dynamic memory. SSN splits a given document into multiple segments, encodes them with BERT (Devlin et al., 2019), and selects salient sentences. Instead of representing the document as a graph, it uses a graph-based memory module, updated iteratively with a GAT (Veličković et al., 2018), to allow information to flow across different windows. HeterGraphLongSum (Phan et al., 2022) utilized words, sentences, and passages as nodes, while considering undirected edges for words in sentences, and directed edges for words in passages and passage to sentences. Instead of using pre-trained embeddings, it used CNNs and bidirectional LSTMs for node encoding, yielding outstanding results. MTGNN-SUM (Doan et al., 2022) achieved similar results by capturing both inter and intra-sentence information when combining a homogeneous graph of sentence nodes with a heterogeneous graph of words and sentences, as in Wang et al. (2020).

Recent studies underscore the importance of structural information in long document summarization. HEGEL (Zhang et al., 2022) represented documents as hypergraphs with hyperedges joining multiple vertices, incorporating semantic connections such as keyword coreference, section structure, and latent topics. CHANGES (Zhang et al., 2023) introduced a sentence–section hierarchical graph, creating fully connected subgraphs for sentences and sections, and linking sentence nodes to their respective section nodes.

Sentence Labeling Most previous work (Jia et al., 2020; Zhang et al., 2022; Wang et al., 2024) adopted the greedy labeling approach from Nallapati et al. (2017) without specifying the used n-gram level for the ROUGE metric. Since ROUGE can be computed for measuring the matching of unigrams, bigrams, or longest common subsequences, different settings can significantly affect the performance of the sentence classifier. Some methods (Wang et al., 2020; Doan et al., 2022; Zhang et al., 2023) followed Liu and Lapata (2019), which selected sentences that maximize the ROUGE-2 score against the gold summary. Other works (Cui et al., 2020; Cui and Hu, 2021; Phan et al., 2022) used pre-labeled benchmarks (Xiao and Carenini, 2019), where labels were assigned by greedily optimizing ROUGE-1. Conversely, Cho et al. (2022) selected sentences that maximize the average of ROUGE-1 and ROUGE-2 F1-scores.

3 GraphLSS

Inspired by previous work, we propose a heterogeneous model using sentences and words as nodes, with four edge types to capture Lexical, Structural, and Semantic features. Our graphs are processed by a heterogeneous GAT (Veličković et al., 2018), followed by a sentence node classifier.

Graph Construction We represent a document as an undirected graph $G = (V, E)$, where the node set is defined as $V = V_{\text{s}} \cup V_{\text{w}}$, and the edge set $E = \{E_{\text{ss}}, E_{\text{ns}}, E_{\text{ws}}, E_{\text{ww}}\}$. Here, V_{s} corresponds to the n sentences in the document, and V_{w} denotes the set of m unique words of the document, limited to the most pertinent ones in terms of semantic richness, nouns and verbs. Conversely, E_{ss} includes sentence pair edges, weighted by cosine similarity, within a predefined window size to account for local similarity and prevent dense graphs. Boolean edges E_{ns} indicate the sentence occurrence order in documents. E_{ws} denotes words in sentences via tf-idf weighted edges, and E_{ww} captures weighted edges for word pairs using cosine similarity.

Extractive Labels There is no consensus on how to effectively generate extractive ground truth labels. We label the data by greedily optimizing the ROUGE-1 score, a simple and intuitive method widely adopted in previous work. This method allows us to label more sentences as relevant compared to other strategies. Instead of using the data published by Xiao and Carenini (2019), we preprocess and label the datasets from scratch.

Adaptive Class Weights Since the extractive ground truth labels for long documents are highly imbalanced, we optimize the GAT model using weighted cross-entropy loss. We assign initial class weights to relevant and irrelevant sentences, employing adaptive class weights for the relevant class and static weights for non-summary sentences as:

$$\lambda^{i+1} = \lambda^{i} - \left(\tau - \frac{\tau}{\log(\tau)}\right), \qquad (1)$$

where τ corresponds to the portion of sentences predicted as relevant for the summary in relation to the total number of existing sentences.

4 Experiments

Datasets We use two publicly available benchmarks for long document summarization, PubMed and arXiv (Cohan et al., 2018). PubMed comprises biomedical scientific papers, while arXiv covers various scientific domains. Both datasets contain English articles, and are widely used by previous work (Table 1). Their statistics and preprocessing details are provided in Appendix A. Our data and code are available on `<anonymized>`.

Comparison Methods For a more detailed comparative analysis with the models that achieved the best benchmark results (Topic-GraphSum, SSN, and HeterGraphLongSum), we also executed our model using the preprocessed data and sentence-level relevance labels provided by Xiao and Carenini (2019). Additionally, we include results from recent non-graph extractive summarizers in Table 1 for reference; Lodoss (Cho et al., 2022) learns sentence representations through simultaneous summarization and section segmentation, Topic-Hierarchical-Sum (Wang et al., 2024) uses local topic information and hierarchical extraction modules, and LOCOST (Le Bronnec et al., 2024) is an abstractive summarization model based on state-space models for conditional text generation.

Experimental Setup We trained a GAT model (Veličković et al., 2018) with 4 attention heads, varying the number of hidden layers between 1 and 2. We applied Dropout after every GAT layer with a retention probability of 0.7. The final representation is fed into a sigmoid classifier. We initialized word nodes using GloVe Wiki-Gigaword 300-dim. embeddings (Pennington et al., 2014) and pre-trained SBERT (`All-MiniLM-L6-v2`) embeddings for sentence nodes (Reimers and Gurevych, 2019). Notably, our word nodes are restricted to the top 50,000 most frequent words in the respective dataset's vocabulary. All experiments used a batch size of 64 samples and were trained for a maximum of 20 epochs using Adam optimization with an initial learning rate of 10^{-3}. The training was stopped if the validation loss did not improve for 7 consecutive iterations. The objective function of each model was to minimize the binary cross-entropy loss using class weights, as described in Equation 1 (more details in Appendix B). All experiments are based on PyTorch Geometric and conducted on an NVIDIA GeForce RTX 3050.

5 Results & Analysis

Table 1 presents the results of different models on both datasets. The first section includes graph-based summarization models, including the Oracle results reported in Xiao and Carenini (2019). The second section includes non-graph summarizers as reference, and the third section includes our results. ROUGE is used as the evaluation metric, including ROUGE-1/-2/-L F1-score for measuring the informativeness and fluency of the summaries.

Summarization Results GraphLSS significantly outperforms all compared approaches in ROUGE-1/-2/-L scores on PubMed and arXiv, showing effectiveness in identifying relevant sentences in highly imbalanced settings (Equation 1). These results are based on our own preprocessing and labeling. Table 1 also shows the Oracle results using our labels, which greatly exceed those achieved with the labels of Xiao and Carenini (2019). Yet, when using those labels, GraphLSS does not achieve the best results, but still remains competitive, particularly in terms of ROUGE-L. This means that the summaries generated by GraphLSS closely match the gold summaries in terms of the longest common subsequence. Such results also suggest that GraphLSS, even when trained over previously labeled data, obtains better results than recently proposed non-graph models. Although other graph methods may show better results, they are included for reference only, as they are not directly comparable due to the use of different sentence labeling strategies in part requiring extrinsic resources.

Preprocessing and Labeling Table 1 shows that ROUGE scores can vary significantly depending not only on the graph construction and model, but also on the strategy used for generating extractive labels. This crucial aspect has been overlooked in related work, which often focuses on ROUGE results without considering whether the corresponding methods are using the same labeling approach. Moreover, preprocessing steps prior to label calculation can also affect the results. Although Xiao and Carenini (2019) and our study both aimed to maximize the ROUGE-1 score, our labels differ significantly. Comparable setups are a requirement to accurately assess the advantages of models.

GraphLSS Learning Table 2 shows that a two-layer heterogeneous GAT yields better results com-

Model	PubMed R-1	PubMed R-2	PubMed R-L	arXiv R-1	arXiv R-2	arXiv R-L
Graph-based Strategies						
Oracle (Xiao and Carenini, 2019)	55.05	27.48	38.66	53.88	23.05	34.90
Topic-GraphSum (Cui et al., 2020) †	**48.85**	<u>21.76</u>	35.19	<u>46.05</u>	**19.97**	33.61
SSN (Cui and Hu, 2021) †	46.73	21.00	34.10	45.03	19.03	32.58
HeterGraphLongSum (Phan et al., 2022) †	**48.86**	**22.63**	**44.19**	**47.36**	<u>19.11</u>	**41.47**
MTGNN-SUM (Doan et al., 2022)	48.42	22.26	43.66	46.39	18.58	40.50
HEGEL (Zhang et al., 2022)	47.13	21.00	42.18	46.41	18.17	39.89
CHANGES (Zhang et al., 2023)	46.43	21.17	41.58	45.61	18.02	40.06
Non-graph Strategies						
Lodoss (Cho et al., 2022)	49.38	23.89	44.84	48.45	20.72	42.55
Topic-Hierarchical-Sum (Wang et al., 2024)	46.49	20.52	42.06	45.84	19.03	40.36
LOCOST (Le Bronnec et al., 2024)	45.70	20.10	42.00	43.80	17.00	39.70
GraphLSS						
- Our Oracle	60.58	36.91	55.32	63.57	30.40	54.10
- GraphLSS + Labels by Xiao and Carenini (2019) †	<u>47.85</u>	21.74	<u>42.22</u>	45.91	18.35	<u>40.07</u>
- GraphLSS + Our labels	⋆51.42	⋆24.32	⋆49.48	⋆55.14	⋆23.00	⋆50.83

Table 1: ROUGE F1 summarization results. Scores are obtained from the respective papers. Models marked with † used sentence-level labels from Xiao and Carenini (2019), making them directly comparable. We highlight the best results in bold and underline the second-best. GraphLSS results are reported by averaging 3 runs.

pared to a single-layer GAT, indicating the advantage of message passing across multiple semantic units in an extended neighborhood. This applies for both datasets. Additionally, previous work has not adequately addressed the balance between precision and recall, focusing solely on reporting the F1 score without analyzing the individual values and their implications. Our results show that precision and recall are similar for the experiments on the PubMed dataset, achieving a good match between generated summaries and gold summaries for both ROUGE-1 and ROUGE-2. In contrast, on the arXiv dataset, the recall is significantly higher than precision, indicating that while our model retrieves valuable information, the generated summaries are contaminated with additional text. This effect is more pronounced when using two layers for the GAT. In such cases, while the precision does not improve compared to using only one GAT layer, the recall increases considerably. This means that more text is correctly retrieved for the summary, but the exactness of these summaries remains unchanged. Interestingly, this discrepancy is not observed when applying GraphLSS to the previously labeled data by Xiao and Carenini (2019), where precision and recall are balanced. This suggests that the observed differences are due to artifacts in the data labeling procedure rather than the graph construction proposed here, or the trained GAT model, emphasizing our earlier discussion.

Dataset	L	ROUGE-1 P	ROUGE-1 R	ROUGE-1 F1	ROUGE-2 P	ROUGE-2 R	ROUGE-2 F1
PubMed	1	49.75	50.00	49.92	22.61	24.71	23.17
	2	52.59	50.11	51.42	23.91	23.82	24.32
	2*	46.43	49.42	47.85	22.42	21.14	21.74
arXiv	1	45.66	66.68	54.23	17.14	30.20	22.31
	2	45.20	71.04	55.14	17.02	35.74	23.00
	2*	44.88	47.04	45.91	19.96	16.99	18.35

Table 2: GraphLSS precision (P) and recall (R) using our labels. L indicates the number of GAT layers used, and the mark * indicates the results obtained by using data from Xiao and Carenini (2019).

6 Conclusions

We introduced GraphLSS, a heterogeneous graph for long document extractive summarization incorporating lexical, structural, and semantic features. Our experiments on PubMed and arXiv datasets highlight the impact of extractive labels due to their inherent imbalance. GraphLSS demonstrates competitiveness with top-performing graph-based methods and outperforms recent non-graph models by employing a greedy labeling strategy and adaptive weights during training. Future work will explore integrating an abstractive summarizer based on our extractive results to potentially enhance summarization outcomes.

Limitations

While we showed the impact and potential of GraphLSS for long document extractive summarization, there are some points to keep in mind.

Storing document graphs as a data structure obtained from the original documents (texts) involves significant additional disk usage. Previous strategies create such structures on the fly while training the underlying GNN models, and others opt for storing such graphs on disk to speed up model training. We follow the latter strategy. Therefore, the training time reported does not consider the creation of the underlying graphs.

Furthermore, our proposal was only validated on English datasets. Applying GraphLSS to other languages may yield significantly different results, since pre-trained word and sentence embeddings are required for node initialization and thus, training the heterogeneous GAT model. Analyzing this aspect would be particularly interesting for low-resource languages. Additionally, our experiments focus on scientific papers. Although they cover multiple scientific domains, exploring other kinds of long document, e.g., narrative and legal documents, is encouraged. Also, additional data collections should be analyzed in order to generalize our findings to broader domains.

Ethics Statement

While extractive summaries are less prone to hallucinated content, in some instances, they may be misleading due to missing context. Another concern is that of possible bias during the content selection. Depending on the graph construction applied, a GAT model may favor certain types of content over others, such as popular sentences and entities with high degrees, as they might receive more attention. Thus, special care must be taken when relying on summaries to make high-stakes decisions, for example in the legal or medical domains.

Summarizing articles often involves extracting information related to trending topics, institutions, people, and other entities. Balancing the delivery of valuable summaries while respecting the privacy of these entities is essential. One strategy to alleviate such concern is anonymization, which ensures that the summary content does not reveal sensitive features. In our study, we conduct all experiments on publicly available scientific articles, and hence have forgone such anonymization.

References

Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward unifying text segmentation and long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 106–118, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Cui and Le Hu. 2021. Sliding selector network with dynamic memory for extractive summarization of long documents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5881–5891, Online. Association for Computational Linguistics.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5360–5371, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuan-Dung Doan, Le-Minh Nguyen, and Khac-Hoai Nam Bui. 2022. Multi graph neural network for extractive long document summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5870–5875, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631, Online. Association for Computational Linguistics.

Florian Le Bronnec, Song Duong, Mathieu Ravaut, Alexandre Allauzen, Nancy Chen, Vincent Guigue, Alberto Lumbreras, Laure Soulier, and Patrick Gallinari. 2024. LOCOST: State-space models for long document abstractive summarization. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1144–1159, St. Julian's, Malta. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Tuan-Anh Phan, Ngoc-Dung Ngoc Nguyen, and Khac-Hoai Nam Bui. 2022. HeterGraphLongSum: Heterogeneous graph neural network with passage aggregation for extractive long document summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6248–6258, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Antoine Tixier, Polykarpos Meladianos, and Michalis Vazirgiannis. 2017. Combining graph degeneracy and submodularity for unsupervised extractive summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 48–58, Copenhagen, Denmark. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 2018 International Conference on Learning Representations (ICLR)*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Ting Wang, Chuan Yang, Maoyang Zou, Jiaying Liang, Dong Xiang, Wenjie Yang, Hongyang Wang, and Jia Li. 2024. A study of extractive summarization of long documents incorporating local topic and hierarchical information. *Scientific Reports*, 14(1):10140.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. 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 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2022. HEGEL: Hypergraph transformer for long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10167–10176, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Contrastive hierarchical discourse graph for scientific document summarization. In *Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023)*, pages 37–47, Toronto, Canada. Association for Computational Linguistics.

A Datasets

For the data prepossessing, we removed instances of empty summaries and instances where the article is shorter than the summarization, and split the documents via NLTK's sentence tokenizer.

Since the sentence tokenizer splits text based on punctuation, this can often result in meaningless generated sentences. For example, the sentence *"Neptune masses can be excluded by our limits determinations (fig.1)"* results in a head sentence S_h =*"Neptune masses can be excluded by our limits determinations (fig."* and a tail sentence S_t =*"1)."*. In such cases, we merged tail sentences with the preceding ones to maintain text coherence.

B Further Experiment Details

Adaptive Class Weights Figure 1 illustrates how the adaptive class weights evolve across epochs during training. Specifically, we update the weights solely for the relevant class (summary sentences), maintaining static weights for the irrelevant class.

	PubMed	arXiv
#Training	115,776	197,650
#Validation	6,584	6,435
#Testing	6,620	6,439
Avg. # Tokens in doc.	2,768	3,913
Avg. # Tokens in summary	205	203
Avg. # Sentences in doc.	89	133
Avg. # Sentences in summary	8	7

Table 3: Datasets statistics.

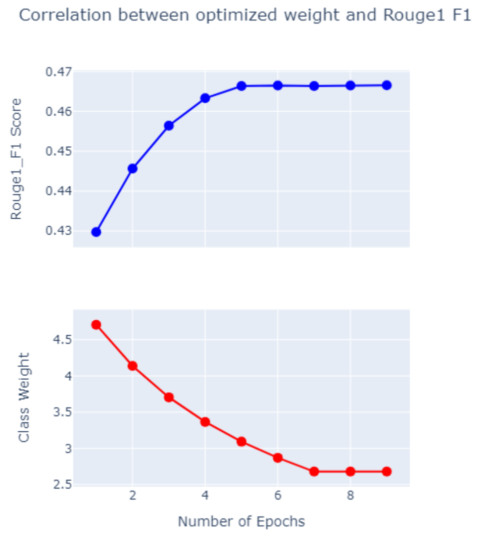

Figure 1: Effect of adaptive class weights on PubMed.

Training Time Table 4 shows the average execution time for GAT training on GraphLSS, using our extractive labels. It also provides the average number of nodes and edges for our constructed document graphs on each dataset.

All experiments are based on PyTorch Geometric and conducted on an NVIDIA GeForce RTX 3050.

	PubMed			arXiv		
L	Nodes	Edges	Time	Nodes	Edges	Time
1	265.4	365.6	1,193 min	299.2	1146.0	1,365 min
2			1,566 min			1,912 min

Table 4: Average execution time for training. L indicates the number of GAT layers used.

Libraries The experiments were conducted using the following libraries:

Library	Version
`nltk`	`3.8.1`
`pytorch`	`2.2.1`
`transformers`	`4.38.2`
`rouge`	`1.0.1`
`scikit-learn`	`1.3.0`
`torchmetrics`	`1.2.1`
`torch_geometric`	`2.5.0`

Table 5: Libraries and versions.