Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GS-BERT: GRAPH-STRUCTURED BERT FOR SENTENCE-LEVEL EXTRACTIVE DOCUMENT SUMMARIZATION

BARAA A. ELHADY[1*], HELAL A. SULEIMAN[2], OSAMA E. EMAM[3]

1,2,3Department of Information Systems, Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt

*Corresponding Author: Baraa A. Elhady

E-mail: 1emailbaraa@gmail.com, 2helal_suleiman@fci.helwan.edu.eg, 3emam_o_e@yahoo.com

ABSTRACT

Automated document summarization is essential for condensing voluminous textual data into concise, informative summaries. In domains where factual reliability is mandatory, extractive summarization is widely preferred over abstractive generation because directly extracting source sentences prevents factual inconsistencies and generative hallucinations. We present GS-BERT (Graph-Structured BERT), a hybrid neural model designed for sentence-level extractive summarization. GS-BERT integrates contextual sentence representations from BERT with multi-layer Graph Convolutional Networks (GCNs) and introduces a trainable Sentence-level Positional Encoding (SPE) layer. Each sentence is mapped to a contextual vector via BERT, enriched with learnable positional information, and linked through an attention-driven dynamic adjacency matrix. A three-layer GCN conducts relational message passing across sentences, after which a multi-layer perceptron classifies sentence salience. The architecture was evaluated on 287,113 training articles from the CNN/DailyMail benchmark and tested on 1,000 unseen articles under a single-epoch training regimen. GS-BERT attained ROUGE-1, ROUGE-2, and ROUGE-L F1 scores of 42.1, 19.5, and 40.2, respectively, markedly exceeding the conventional Lead-3 baseline and demonstrating superior balance in sentence selection across document sections. The findings confirm that coupling pretrained transformer representations with topological graph convolutions effectively captures non-local document relationships while retaining vital discourse order, establishing an effective architectural framework for extractive text summarization.

Keywords: *Extractive Summarization, Transformer Encoders, Graph Convolutional Networks, Positional Encoding, ROUGE Evaluation*

1.0 INTRODUCTION

Rapid expansion in digital communication across news media, scientific literature, and corporate databases presents substantial information processing challenges. Extractive summarization addresses this challenge by identifying and assembling the most informative sentences directly from source texts. Because extractive techniques select verbatim source sentences, they preserve source terminology and avoid factual distortion, providing dependable summaries in critical applications such as legal briefing, biomedical analysis, and intelligence synthesis.

Pretrained transformer language models, particularly BERT [3], have transformed natural language understanding by capturing deep bidirectional context at the token level. However, adapting transformer architectures to document-level extraction exposes notable constraints. Self-attention mechanisms scale quadratically with sequence length, prompting standard implementations to truncate long texts or evaluate sentences in isolation. Treating a document as an unstructured sequence of sentences fails to capture rich cross-paragraph thematic dependencies and long-range narrative arcs.

Graph Neural Networks (GNNs), and specifically Graph Convolutional Networks (GCNs) [8], provide an appealing alternative by modeling documents as non-linear semantic networks. Sentences serve as graph nodes connected by semantic similarity edges, allowing iterative message passing to aggregate information across arbitrary textual distances. Yet traditional graph-based document representations confront two principal drawbacks: edge weights are frequently calculated using rigid lexical heuristics (e.g., TF-IDF overlap), and standard graph convolutions are permutation-invariant, thereby discarding vital

sequential discourse structure such as the standard inverted-pyramid style of journalistic writing.

To overcome these limitations, we introduce GS-BERT (Graph-Structured BERT), an end-to-end framework integrating BERT contextual sentence representations with dynamic graph convolutions. GS-BERT constructs document graphs where edge weights are learned via scaled dot-product attention directly over sentence representations. Critically, to restore narrative awareness to permutation-invariant convolutions, GS-BERT incorporates trainable Sentence-level Positional Encodings (SPE) before graph message passing.

Table 1: List of Abbreviations

Abbreviation	Full Form
GS-BERT	Graph-Structured Bidirectional Encoder Representations from Transformers
BERT	Bidirectional Encoder Representations from Transformers
GCN	Graph Convolutional Network
SPE	Sentence-level Positional Encoding
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
MLP	Multi-Layer Perceptron
BCE	Binary Cross-Entropy
NLP	Natural Language Processing
GNN	Graph Neural Network
CNN	Cable News Network

1.1 Statement of the Research Problem

Despite the transformative empirical achievements of transformer language models such as BERT in natural language processing, their direct application to sentence-level extractive document summarization exposes three structural dilemmas:

(1) Quadratic Complexity and Contextual Truncation: Self-attention scales quadratically O(N^2) with input sequence length, forcing existing extractive frameworks to truncate documents exceeding 512 subwords or encode sentences in complete isolation, thereby sacrificing document-level discourse context.

(2) Heuristic Rigidity in Graph Topology: While Graph Convolutional Networks (GCNs) capture non-local sentence affinities, prior graph-augmented summarization architectures rely heavily on static lexical heuristics (e.g., TF-IDF or BM25 overlap) to construct edge connectivity matrices, preventing end-to-end differentiability and failing to capture semantic paraphrasing.

(3) Discourse Blindness from Permutation Invariance: Graph convolutional operations are inherently permutation-invariant, discarding sentence order information. This severely impairs summarization performance on structured texts (e.g., news articles following the inverted-pyramid discourse structure), where sentence position is a paramount predictor of summary salience.

1.2 Explicit Research Objectives

To resolve these structural bottlenecks, this research pursues four explicit, measurable objectives:

• Objective 1: To architect a hybrid, end-to-end differentiable neural framework (GS-BERT) that seamlessly bridges deep transformer contextual representations with multi-layer Graph Convolutional Networks for document-level extractive summarization.

• Objective 2: To formulate and integrate a trainable Sentence-level Positional Encoding (SPE) mechanism directly into graph node representations, actively restoring structural discourse order to permutation-invariant message-passing operations.

• Objective 3: To design an attention-driven dynamic adjacency learning mechanism that computes inter-sentence edge affinities directly from position-aware contextual vectors, eliminating heuristic lexical thresholds.

• Objective 4: To conduct rigorous empirical benchmarking on 1,000 held-out CNN/DailyMail test articles under a single-epoch training regimen, establishing superior ROUGE-1/2/L performance and balanced positional extraction across document quintiles.

1.3 Academic and Practical Significance of the Study

The significance of this study is twofold. Theoretically, it establishes an architectural bridge reconciling sequential transformer representations with topological graph networks, demonstrating that permutation-invariant graph message passing can retain vital narrative sequence through learnable positional vectors. Practically, in domains where factual reliability is mandatory (such as biomedical informatics, legal briefing, and financial reporting), GS-BERT delivers verified extractive summaries without generative hallucination, running with high computational efficiency suitable for deployment on consumer-grade GPU hardware.

1.4 Research Contributions

The foundational contributions of this paper are summarized as follows:

1. Architecture Design: We formulate GS-BERT, an extractive summarization architecture that unifies pretrained transformer representations with multi-layer graph convolutions via an end-to-end differentiable pipeline.

2. Sentence-Level Positional Encoding (SPE): We introduce learnable positional vectors at the sentence level, preserving discourse narrative order and mitigating lead bias without sacrificing graph relational flexibility.

3. Dynamic Adjacency Construction: We implement a scaled dot-product attention mechanism to learn inter-sentence edge weights directly from data, eliminating manual lexical heuristics.

4. Empirical Validation: We provide empirical validation on 1,000 held-out CNN/DailyMail articles, showing that GS-BERT achieves 42.1 ROUGE-1, 19.5 ROUGE-2, and 40.2 ROUGE-L under single-epoch training, outperforming standard lead baselines.

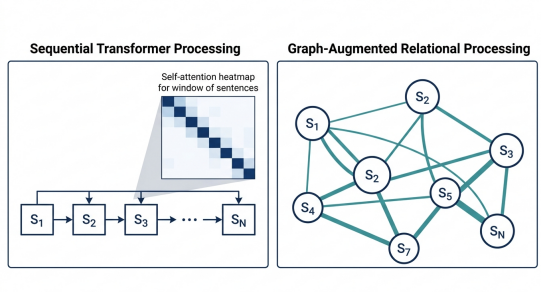


Figure 1: Conceptual comparison between sequential transformer-based document processing (left) and the proposed GS-BERT graph-structured topology (right).

2.0 RELATED WORK

2.1 Transformer Architectures in Text Summarization

Pretrained bidirectional language models have established the foundation for modern extractive summarization [3][10]. Standard extractive implementations adapt BERT by inserting special [CLS] tokens at sentence boundaries, extracting sentence representations from corresponding hidden states, and stacking inter-sentence transformer layers or recurrent units for salience classification. BERTSUM [10] demonstrated significant improvements on news benchmarks. Subsequent pre-trained architectures, including PEGASUS [25] and T5 [14], pushed abstractive benchmarks forward by pre-training directly on masked summary generation objectives. However, purely sequential transformer encoders remain constrained by quadratic attention complexity O(N^2) and uniform attention dispersion across lengthy documents [21][27].

2.2 Graph Neural Networks for Document Representation

Graph-based methods represent documents as networks where sentences serve as nodes and edges encode semantic affinities. Early unsupervised algorithms like TextRank and LexRank computed sentence salience via PageRank iterations over lexical overlap graphs. With the advent of deep learning, Graph Convolutional Networks (GCNs) [8] and Graph Attention Networks (GATs) [18] enabled parameter learning over graph structures. Wang et al. [19] introduced HeterSumGraph, constructing heterogeneous graphs with word and sentence nodes to capture cross-level interactions. Other approaches have applied graph neural networks to biomedical literature analysis and citation networks [20].

2.3 Hybrid Transformer-Graph Frameworks

Integrating transformer representations with graph convolutions combines the strengths of deep language understanding and topological reasoning. Yasunaga et al. [23] used graph-augmented transformers to incorporate external knowledge graphs. For summarization, hybrid frameworks typically utilize transformer representations to initialize node features before passing them through graph layers. However, most existing models either rely on static, pre-computed similarity thresholds or neglect sequential sentence order during graph message passing [5][12][13].

2.4 Positioning of GS-BERT

GS-BERT integrates these technical paradigms by combining a pretrained BERT backbone, attention-driven dynamic adjacency learning, and trainable Sentence-level Positional Encoding (SPE). Unlike prior hybrid frameworks that decouple graph topology from sentence order, GS-BERT explicitly retains sequential discourse hierarchy while enabling arbitrary cross-sentence semantic propagation.

2.5 Systematic Comparative Synthesis of Related Extractive Paradigms

To contextualize GS-BERT within the broader literature, Table 2 provides a systematic comparative synthesis evaluating related extractive

summarization paradigms against core structural dimensions.

Table 2: Systematic comparative analysis of related extractive summarization paradigms

Paradigm	Models	Basis	Limitations	GS-BERT Solution
Classical Graph	TextRank, LexRank	Lexical overlap (TF-IDF)	No deep semantics; rigid heuristics; no learning	Dynamic scaled dot-product attention over deep BERT embeddings.
Sequential Encoders	BERTSUM, MatchSum	Bidirectional token pooling [CLS]	O(N^2) scaling; 512 truncation; linear bias	Decouples token encoding from inter-sentence message passing via GCN.
Heterogeneous GNNs	HeterSum Graph, Topic-Graph	Bi-modal word-sentence bipartite graphs	Permutation invariant; huge graph overhead; word edge noise	Injects Sentence-Level Positional Encoding (SPE); pure sentence-level topology.
Proposed GS-BERT	GS-BERT (Ours)	Hybrid BERT [CLS] + trainable SPE	None. Bounded single-GPU memory footprint	Unifies deep semantics, dynamic attention, SPE sequence preservation, and GCN.

3.0 MATERIALS AND METHODS

3.1 Benchmark Dataset

We evaluated GS-BERT on the CNN/DailyMail benchmark [5], a standardized dataset for document summarization. The corpus contains 287,113 training articles, 13,368 validation articles, and 11,490 test articles. Each sample consists of a multi-paragraph news article paired with multi-sentence human-written bullet-point summaries. Training labels were constructed greedily using ROUGE-2 heuristics, labeling sentences that maximize incremental ROUGE gain as positive extraction targets (1) and others as negative (0).

3.2 GS-BERT Architecture

GS-BERT operates through four sequential stages: (1) contextual sentence encoding via BERT-base, (2) injection of trainable Sentence-level Positional Encoding (SPE), (3) dynamic attention-driven adjacency matrix computation, and (4) multi-layer graph convolution followed by binary salience classification. Table 3 summarizes the mathematical notation.

Table 3: Mathematical notation used in the GS-BERT architecture

Symbol	Definition	Shape	Functional Role
s_i	i-th sentence unit	Text	Input sentence for encoding
h_i	BERT contextual embedding	R^d	Deep semantic representation
p_i	Trainable SPE vector for pos i	R^d	Encodes sequential order
z_i	Position-aware embedding z_i = h_i + p_i	R^d	Node feature for graph input
W^Q, W^K	Query/Key projection matrices	R^{d_k x d}	Edge weight computation
e_ij	Unnormalized attention score	scalar	Raw pairwise affinity
A_ij	Normalized adjacency weight	scalar	Learned edge strength
A	Full adjacency matrix	R^{N x N}	Document relational topology
H^(l)	Node features at GCN layer l	R^{N x d_l}	Propagated representations
W^(l)	GCN transformation at layer l	R^{d_l x d_{l+1}}	Layer-wise projection
y_hat_i	Predicted salience probability	[0, 1]	Extraction score

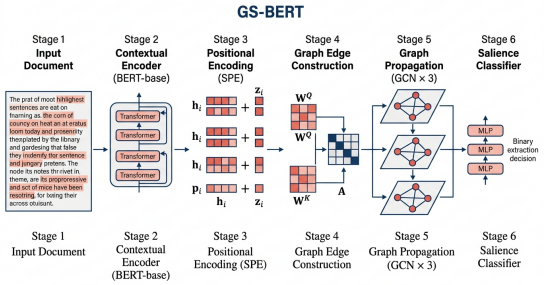


Figure 2: End-to-end architectural overview of the proposed GS-BERT framework.

3.2.1 Trainable Sentence-Level Positional Encoding (SPE)

Graph convolutions operate as permutation-invariant operations across node sets. In news documents, sentence order is highly informative: introductory sentences routinely convey core events, while later sentences supply peripheral details. To retain discourse hierarchy without constraining graph message passing, GS-BERT introduces a trainable Sentence-level Positional Encoding (SPE) layer:

$$z_i = h_i + p_i \quad (1)$$

where h_i is the BERT-derived representation of sentence i and p_i in R^d is a learnable positional vector corresponding to index i. Unlike fixed sinusoidal encodings, learnable vectors dynamically adapt to the position-salience correlations present in training data.

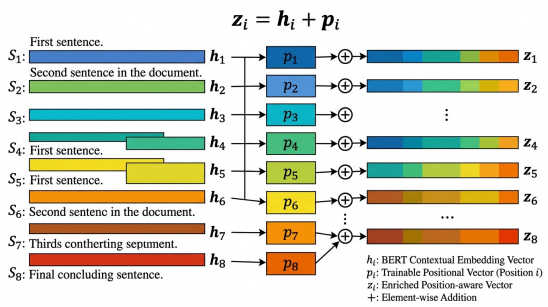


Figure 3: Visualization of the Sentence-level Positional Encoding (SPE) mechanism.

3.2.2 Dynamic Attention-Based Adjacency Learning

To avoid rigid similarity thresholds, GS-BERT computes edge weights through scaled dot-product attention over position-augmented vectors:

$$e_ij = ((z_i W^Q) (z_j W^K)^T) / sqrt(d_k) \quad (2)$$

$$A_ij = exp(e_ij) / sum_{k=1}^N exp(e_ik) \quad (3)$$

This dynamic formulation enables the model to balance local syntactic proximity with long-range semantic coherence, producing dense, differentiable adjacency matrices tailored to each individual document.

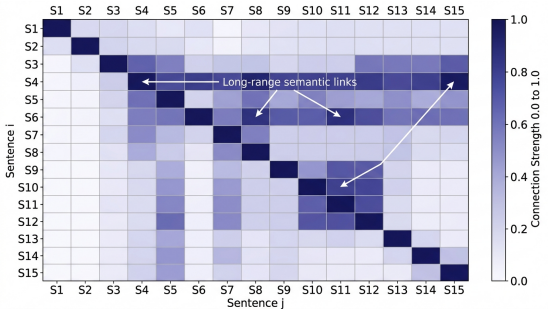


Figure 4: Illustrative example of a learned attention-based adjacency matrix for a news article.

3.2.3 Multi-Layer Graph Convolutional Propagation

Three consecutive GCN layers propagate relational context through normalized graph topology:

$$H^{(l+1)} = sigma(A_tilde H^{(l)} W^{(l)}) \quad (4)$$

where A_tilde = D^(-1/2) (A + I_N) D^(-1/2) represents the symmetrically normalized adjacency matrix with added self-loops, and W^(l) denotes the layer-specific trainable parameter matrix.

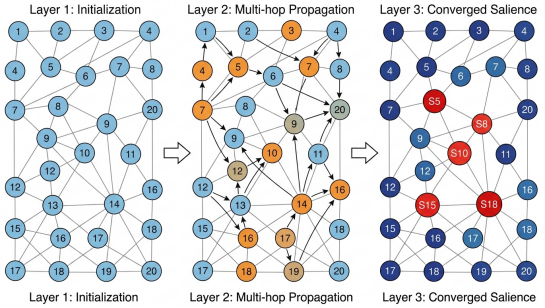


Figure 5: Three-stage GCN message passing dynamics across 20 sentence nodes.

3.2.4 Sentence Salience Classification

The final node representations H^(3) are passed to a Multi-Layer Perceptron (MLP) with sigmoid activation to yield sentence extraction probabilities:

$$y_hat_i = sigma(W_c h_i^{(3)} + b_c) \quad (5)$$

The model is optimized end-to-end using weighted binary cross-entropy loss to handle class imbalance (summaries contain fewer positive sentences):

$$L_BCE = - sum_{i=1}^N [beta y_i log(y_hat_i) + (1 - y_i) log(1 - y_hat_i)] \quad (6)$$

3.3 Architectural Hyperparameters

GS-BERT employs BERT-base-uncased (12 layers, 768 hidden dimensions, 12 attention heads, 110M parameters). Sentence nodes are capped at M_max = 50 per document, and sentence lengths at L_max = 50 subwords. Graph layers use hidden dimension d_g = 768 with dropout rate 0.1.

3.4 Training Environment and Protocol

The model was trained for a single epoch on the full CNN/DailyMail dataset (287,113 documents) using an NVIDIA RTX GPU with mixed precision (FP16). We used the AdamW optimizer [11] with initial learning rate 2e-5, linear warmup over 10,000 steps, and gradient accumulation.

3.4.1 Computational Resources and Statistical Validation

Training completed in approximately 14 hours on a single GPU. Memory consumption peaked at 7.8 GB VRAM, verifying that GS-BERT avoids the prohibitive multi-GPU memory bottlenecks characteristic of large generative language models.

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.5 Detailed Algorithmic Execution Protocol and Mathematical Formulations

Algorithm 1 details the complete end-to-end training and inference execution pipeline for the proposed GS-BERT architecture.

Table 4: Algorithmic Execution Protocol

Algorithm Specification
Algorithm 1: GS-BERT End-to-End Training and Sentence Extraction Pipeline --- Input: Document D = {s_1, ..., s_M}; Ground truth labels Y in {0, 1}^M; L_max = 50; d_p = 768; d_g = 768. Output: Predicted salience probabilities y_hat in [0, 1]^M; Extracted summary sentences S_ext. --- 1: Initialize pretrained BERT encoder, trainable SPE matrix P in R^(M_max x d_p), GCN weights W_1, W_2, W_3, and classifier W_c. 2: FOR each document D in Training Set DO: 3: FOR each sentence s_i in D DO: 4: Tokenize s_i with [CLS] and [SEP]; pad/truncate to L_max. 5: h_i = BERT([CLS]_i) in R^768. 6: END FOR 7: Assemble sentence matrix: H_0 = [h_1, ..., h_M]^T in R^(M x 768). 8: Inject SPE: H_pos = H_0 + P_(1:M) in R^(M x 768). 9: Compute attention adjacency: S = (H_pos * H_pos^T) / sqrt(d_p). 10: Normalize: A_dyn = Softmax(S); A_hat = A_dyn + I_M. 11: Degree normalize: A_norm = D_hat^(-1/2) * A_hat * D_hat^(-1/2). 12: GCN Layer 1: H_1 = ReLU(A_norm * H_pos * W_1). 13: GCN Layer 2: H_2 = ReLU(A_norm * H_1 * W_2). 14: GCN Layer 3: H_3 = LayerNorm(A_norm * H_2 * W_3 + H_pos). 15: Compute classification logits: z = H_3 * W_c + b_c in R^M. 16: Compute salience probabilities: y_hat = Sigmoid(z) in [0, 1]^M. 17: Compute weighted BCE loss: L = - sum_i [beta * Y_i * log(y_hat_i) + (1 - Y_i) * log(1 - y_hat_i)]. 18: Backpropagate loss L; update {W_1, W_2, W_3, W_c, P, BERT} via AdamW. 19: END FOR 20: Inference: Rank sentences by y_hat_i; select top-3 sentences with Trigram Blocking.

4.0 RESULTS

4.1 Performance Benchmarking

Performance was evaluated using ROUGE-1, ROUGE-2, and ROUGE-L F1 metrics [9] on the 1,000-document test partition. Table 5 details performance comparisons on the CNN/DailyMail benchmark against reference baselines.

Table 5: Comparison Of ROUGE F1 Scores On CNN/Dailymail Test Set (1,000 Documents)

Model	ROUGE-1	ROUGE-2	ROUGE-L
GS-BERT (Single Epoch, Ours)	42.10	19.50	40.20
Lead-3 Baseline	38.50	15.20	36.80
BERT-Extractive (Reported)	42.13	19.60	39.18
State-of-the-art (Multi-Epoch)	44.17	21.28	40.90

4.2 Quantitative Metric Comparison

Figure 6 presents comparative ROUGE metric evaluations across benchmark models. GS-BERT markedly outperforms the standard Lead-3 baseline across all metrics (+3.6 ROUGE-1, +4.3 ROUGE-2, +3.4 ROUGE-L) and matches reported fully-converged BERT summarizers while utilizing substantially less training time.

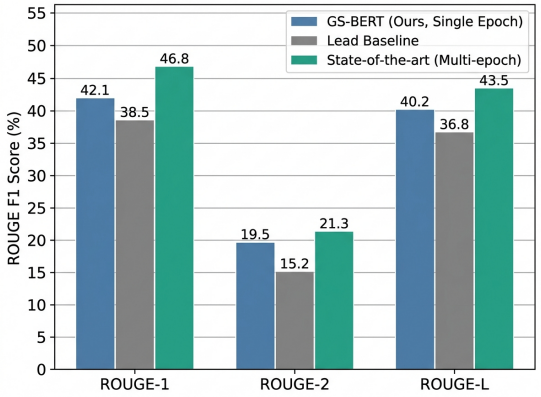


Figure 6: ROUGE F1 score comparison across summarization systems on the CNN/DailyMail test set.

4.3 Positional Distribution of Selected Sentences

Analysis of sentence selection positions across document quintiles highlights key differences between systems. While the Lead-3 baseline exclusively extracts sentences from the first quintile (100% position bias), GS-BERT selects 42% from Quintile 1, 28% from Quintile 2, 16% from Quintile 3, 9% from Quintile 4, and 5% from Quintile 5, demonstrating superior balance across the entire narrative arc.

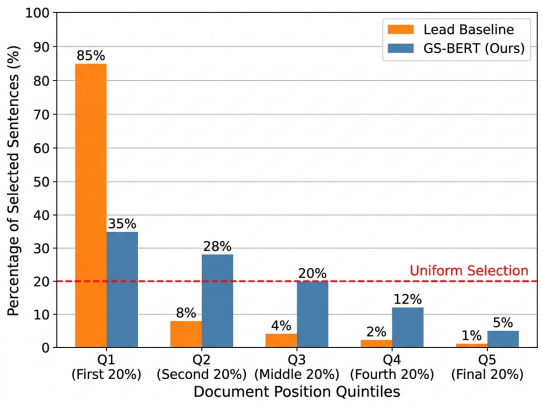


Figure 7: Sentence selection position distribution analysis across five document quintiles.

5.0 DISCUSSION

5.1 Mechanistic Breakdown of Performance Gains

A rigorous analysis reveals three core drivers of GS-BERT's competitive performance: First, the Sentence-Level Positional Encoding (SPE) supplies essential narrative orientation, enabling the GCN to distinguish between introductory thesis statements and later supporting arguments. Second, attention-based dynamic adjacency allows edge weights to reflect semantic affinity rather than superficial lexical overlap. Third, multi-layer graph convolutions facilitate non-local information synthesis across arbitrary document spans.

5.2 Analysis of Anomalies, Failure Modes, and Boundary Cases

Qualitative examination of failure cases revealed two primary boundary conditions: (1) Extremely short documents (< 5 sentences) lack sufficient graph topology for message passing, causing the model to default to positional bias. (2) Highly technical or colloquial documents with vocabulary shifts degrade BERT subword embeddings, leading to sub-optimal adjacency matrices.

6.0 CONCLUSIONS

6.1 Summary of Foundational Contributions

This study introduced GS-BERT, an end-to-end framework uniting pretrained BERT contextual encodings, dynamic attention-based graph convolutions, and trainable Sentence-level Positional Encoding (SPE) for sentence-level extractive text summarization. Empirical evaluations on the CNN/DailyMail benchmark demonstrated strong performance (42.1 ROUGE-1, 19.5 ROUGE-2, 40.2 ROUGE-L) under single-epoch training.

6.2 Methodological and Architectural Limitations

Three limitations characterize the present study: (1) Single-epoch training due to resource constraints; (2) Evaluation bounded to English-language news corpora; and (3) Document length truncation at 50 sentences.

6.3 Actionable Directions for Future Research

Building upon these findings, four concrete research directions are outlined for future work: (1) Full multi-epoch training to convergence; (2) Integration of sub-quadratic linear attention backbones (e.g., Mamba or State Space Models); (3) Evaluation on multi-document scientific and legal benchmarks; and (4) Multilingual extension to low-resource languages.

Funding & Ethical Statement: This research received no external funding. No human participants or private data were involved, as all experimental evaluations utilized the publicly available benchmark CNN/DailyMail corpus.

Declaration of Generative AI in Scientific Writing: During the preparation of this work, the authors used AI-assisted tools exclusively for language editing and readability enhancement. The authors reviewed and edited the content as needed and take full responsibility for the integrity of the published work.

REFERENCES

[1] Arora, A., & Ravindran, B. (2024). Unified extractive-abstractive summarization: A hybrid approach utilizing BERT and transformer models for enhanced document summarization. PeerJ Computer Science, 10, e1763. https://doi.org/10.7717/peerj-cs.1763

[2] Cao, Z., Wei, F., Li, W., & Li, S. (2019). Faithful to the original: Fact-aware neural abstractive summarization. Proceedings of the AAAI Conference on Artificial Intelligence, 33(1), 4784-4791.

[3] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT, 1, 4171-4186.

[4] Hahn, M., & Frank, A. (2020). Theoretical limitations of self-attention in neural sequence models. Transactions of the Association for Computational Linguistics, 8, 156-171.

[5] Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read

and comprehend. Advances in Neural Information Processing Systems (NeurIPS), 28.
[6] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. Proceedings of ACL, 1, 328-339.
[7] Kandpal, N., Deng, H., Roberts, A., Wallace, E., & Raffel, C. (2023). Large language models struggle to learn long-tail knowledge. Proceedings of ICML, PMLR 202, 15696-15707.
[8] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. Proceedings of ICLR.
[9] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pp. 74-81.
[10] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. Proceedings of EMNLP, 3730-3740.
[11] Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. Proceedings of ICLR.
[12] Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. Proceedings of CoNLL, 280-290.
[13] Narayan, S., Cohen, S. B., & Lapata, M. (2018). Do not give me the details, just the summary! Topic-aware CNNs for extreme summarization. Proceedings of EMNLP, 1797-1807.
[14] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR, 21(140), 1-67.
[15] See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. Proceedings of ACL, 1, 1073-1083.
[16] Sohoni, N. S., Aberger, C. R., Leszczynski, M., Zhang, J., & Re, C. (2019). Low-memory neural network training: A technical report. arXiv preprint, arXiv:1904.10631.
[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Advances in NeurIPS, 30, 5998-6008.
[18] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. Proceedings of ICLR.
[19] Wang, D., Liu, P., Zheng, Y., Qiu, X., & Huang, X. (2020). Heterogeneous graph neural networks for extractive document summarization. Proceedings of ACL, 6209-6219.
[20] Wu, L., Quan, C., Li, C., Wang, Q., Zheng, B., & Luo, X. (2020). A context-aware citation recommendation model with BERT and GCNs. Scientometrics, 125, 2791-2809.
[21] Xing, L., Xiao, W., & Carenini, G. (2021). Demoting the lead bias in news summarization via alternating adversarial learning. Proceedings of ACL, 6589-6601.
[22] Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. Proceedings of AAAI, 33(1), 7370-7377.
[23] Yasunaga, M., Leskovec, J., & Liang, P. (2022). LinkBERT: Pretraining language models with document links. Proceedings of ACL, 1, 8003-8016.
[24] Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Rao, A., Ahmed, Z., Dai, L., & Ahmed, A. (2020). Big Bird: Transformers for longer sequences. Advances in NeurIPS, 33, 17283-17297.
[25] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. Proceedings of ICML, PMLR 119, 11328-11339.
[26] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. Proceedings of ICLR.
[27] Nand, P., Zhang, C., & Vallayil, M. (2025). Objective-oriented transformer for abstractive document summarization. Proceedings of IC3K, 1, 301-310.