Received: 12 February 2026 | Revised: 8 June 2026 | Accepted: 23 June 2026 | Published online: 29 July 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–12
DOI: 10.47852/bonviewAIA62029367

BON VIEW PUBLISHING

Efficient Extractive Text Summarization Using BiLSTM, Hypergraph, and Dominating Set Property

Pradeepa Sampath[1], Shrijaa Venkatasubramanian Subashini[2], Vimal Shanmuganathan[3] and Seifedine Kadry[4,*]

[1] *Department of Information Technology, SASTRA Deemed University, India*

[2] *Department of Computer Science and Engineering, SASTRA Deemed University, India*

[3] *Department of Computer Science and Engineering, Kalaignar Karunanidhi Institute of Technology, India*

[4] *Department of Computer Science and Mathematics, Lebanese American University, Lebanon*

Abstract: Automatic text summarization condenses voluminous text while preserving crucial information. Despite significant advancements in this field, challenges persist in accurately identifying key content, maintaining contextual coherence, and ensuring computational efficiency. The existing approaches struggle to effectively capture global relationships between sentences and minimize redundancy. In this study, we propose an extractive text summarization framework using Bidirectional Long Short-Term Memory with a hypergraph and a dominating set mechanism. A neural encoding module derives context-aware sentence embeddings, and a hypergraph-based representation models higher-order relational dependencies among sentences. A dominating set-based selection strategy is then applied to identify the most informative sentences for summary generation. The proposed model is evaluated using the CNN/Daily Mail dataset and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics. Empirical evaluation attains ROUGE-1, ROUGE-2, and ROUGE-L scores of 0.4221, 0.3563, and 0.3901, respectively, and demonstrates the proposed model's capability to effectively capture key content and generate coherent and informative summaries. The proposed framework proves to be an efficient approach to automatic summarization and has the potential to be applied in journalism, healthcare, legal analysis, and digital content management.

Keywords: extractive text summarization, CNN/Daily Mail dataset, BiLSTM, hypergraph, dominating set

1. Introduction

A vast amount of textual data is produced every day in today's digital era by various platforms, including social media, news sources, and academic libraries [1]. In this scenario, automatic text summarization is seen as a vital tool, enabling readers to quickly review lengthy resources [2–4]. Abstractive and extractive summarization are the two main categories of text summarization [5–7]. The former generates concise and coherent summaries by paraphrasing and reformulating the source content, whereas the latter identifies and aggregates key sentences from the original document while preserving lexical and syntactic structure. Extractive summarization preserves factual accuracy, ensures reliability, requires less computation, and maintains original wording, making it preferable in accuracy-critical applications. Recurrent Neural Networks (RNNs) [8–10] and Long Short-Term Memory Networks (LSTMs) [11–13] have proven to be more efficient than conventional machine learning techniques in capturing semantic relationships and long-range dependencies in text. Graph-based techniques like heterogeneous graph neural networks have gained popularity due to their capacity to represent intricate sentence associations, which enhances the summarizing process [14]. Despite their noteworthy accomplishments, current methods like SummaRuNNer [15] and heterogeneous graph-based models struggle to capture inter-sentence relationships, resulting in redundant and incoherent summaries, emphasizing the need for improved summarization that fuses sophisticated feature extraction with graph-based models. This study suggests a novel extractive text summarization framework to address the constraints in the state-of-the-art algorithms by incorporating the following innovations:

1) Preprocessing pipeline: Development of a reliable pipeline for cleaning and tokenizing text obtained from the CNN/Daily Mail dataset to ensure high-quality text input for subsequent processing.
2) Bidirectional Long Short-Term Memory (BiLSTM): The essential features are extracted using the BiLSTM architecture

***Corresponding author:** Seifedine Kadry, Department of Computer Science and Mathematics, Lebanese American University, Lebanon. Email: seifedine.kadry@lau.edu.lb

while retaining long-range relationships and contextual knowledge.

3) Hypergraph-based sentence selection: Using the dominating set property and sentence score, a hypergraph represents sentence relationships and chooses the most pertinent and instructive sentences for summary extraction.

By integrating the above features, the proposed framework overcomes the key limitations of existing extractive methods. Section 2 surveys the existing extractive and graph-based summarization methods. Section 3 describes the proposed technique in detail. Section 4 discusses the experiment results and compares various methods, and Section 5 concludes the ongoing effort and outlines future scope.

2. Literature Review

Recent advancements in extractive text summarization have produced a significant increase in the quality and efficiency of text summaries. A number of studies used transformer-based models to improve summarization performance. Huang [16] advanced a multi-path feature learning approach for extractive summarization, using the CNN/Daily Mail dataset. Their approach captured complete text representation by processing Bidirectional Encoder Representations from Transformers (BERT) outputs through multiple pathways. Experimental results achieved greater ROUGE-1 scores than several baseline models, such as Transformer and BertSum.

Du et al. [17] optimized sentence extraction in large-scale datasets using reinforcement learning techniques. Their new model, when tested using the CNN/Daily Mail dataset, yielded better Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores than the earlier methods. However, the approach's effectiveness was limited to lengthy document summarization, and its generalization to shorter documents remains unclear. Fan et al. [18] proposed the Multi-Features Maximal Marginal Relevance-BERT (MFMMR-BertSum) model for text summarization in the CNN/Daily Mail dataset. The model outperformed other conventional extractive summarizing techniques, including statistical methods and deep learning models. González et al. [19] tested hierarchical neural networks on CNN/Daily Mail and NewsRoom Corpora. Wang et al. [20] used Deep Q-Networks (DQN) for extractive summarization. They considered summarization as a sentence-ranking problem in their DQN-based model and rated sentences according to their importance. Their model opened new directions for reinforcement learning applications in text summarizing and outperformed conventional extractive summarization techniques like transformer-based systems. Rahman and Borah [21] proposed a novel summarization method called NeRoBERTa, which included a nested tree structure that leveraged discourse and used syntactic trees to enhance sentence representation. Their model improved coherence in text summaries when compared to other baseline models. Despite this improvement, the authors acknowledged that BERT-based encoders cannot depict sentence-level information coherently [22]. Wang and Zhuang [23] enhanced their model's denoising capacity and improved the quality of the generated summary by modeling realistic textual transformations. They used contrastive learning to distinguish candidate summaries and original articles and the candidate summaries themselves. Gao et al. [24] proposed a jointly trained text summarization framework that integrated both extractive and abstractive methodologies in a unified architecture. By combining the benefits of extractive models, they achieved better ROUGE scores. They found that abstractive models created a more effective summarization system by providing more fluid summaries. Asmitha et al. [25] examined the abstract generation using four models including GPT-2, T5, BART, and PEGASUS. Their study assessed multiple models with SacreBLEU and ROUGE evaluation metrics to quantify their effectiveness in generating accurate and informative summaries. Huang et al. [26] proposed the Genetic Algorithm-assisted Graph Neural Network (GA-GNN) model for text summarization. The new model outperformed the existing methods by 1.48, 1.26, and 0.44 for ROUGE-1, ROUGE-2, and ROUGE-L measures, respectively. Their studies underscored the progressive advancement of extractive summarization and highlighted the adoption of modern techniques like deep learning architectures, attention mechanisms, reinforcement learning strategies, and hierarchical modeling approaches. Most of the existing works have primarily been evaluated on standard benchmark datasets like CNN/Daily Mail. Adeniyi et al. [27] proposed a text rank algorithm for extractive summarization using a bidirectional RNN to implement their new abstractive text summarizer. Vo et al. [28] proposed an interpretable extractive text summarization framework that combines meta-learning with Bi-LSTM to improve sentence selection accuracy while enhancing model adaptability and explainability across diverse text domains.

Research gap: The proposed text summarization model is a hybrid framework that combines BiLSTM-based semantic feature extraction with hypergraph modeling to capture higher-order sentence relationships. By applying the dominating set principle, it ensures optimal sentence selection and overcomes the limitations of traditional graph-based and sequence-based methods while handling coverage and redundancy.

3. Methodology

The proposed model has four successive stages: (1) data collection and preprocessing, (2) feature extraction with a BiLSTM network, (3) hypergraph construction followed by the application of the dominating set property, and (4) computation of sentence scores for text summarization. The overall architecture of the model is shown in Figure 1.

3.1. Data description

The CNN/Daily Mail dataset from Kaggle can be used as the standard benchmark for summarization tasks. It contains lengthy news articles along with their human-generated summaries, and hence this dataset proves to be ideal for developing, validating, and testing machine learning algorithms. This dataset consists of 11,490 test samples, 13,368 validation samples, and 287,113 training samples. Figure 2 depicts the distribution of data.[1]

3.2. Feature extraction using BiLSTM

The BiLSTM network is used for feature extraction. In the forward phase, the network processes the sentence from left to right, whereas in the backward phase, it processes it from right to left, thus allowing the network to extract information in both directions and capture all textual dependencies [29–31]. Figure 3 illustrates BiLSTM feature extraction.

Embedding Layer maps a discrete sentence to vector spaces. This process results in an embedding layer presented as a matrix. This can be shown mathematically using Equation (1) below, where V represents the vocabulary size, while d represents the

[1] https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail/data1

Figure 1
Architecture of the proposed text summarization model

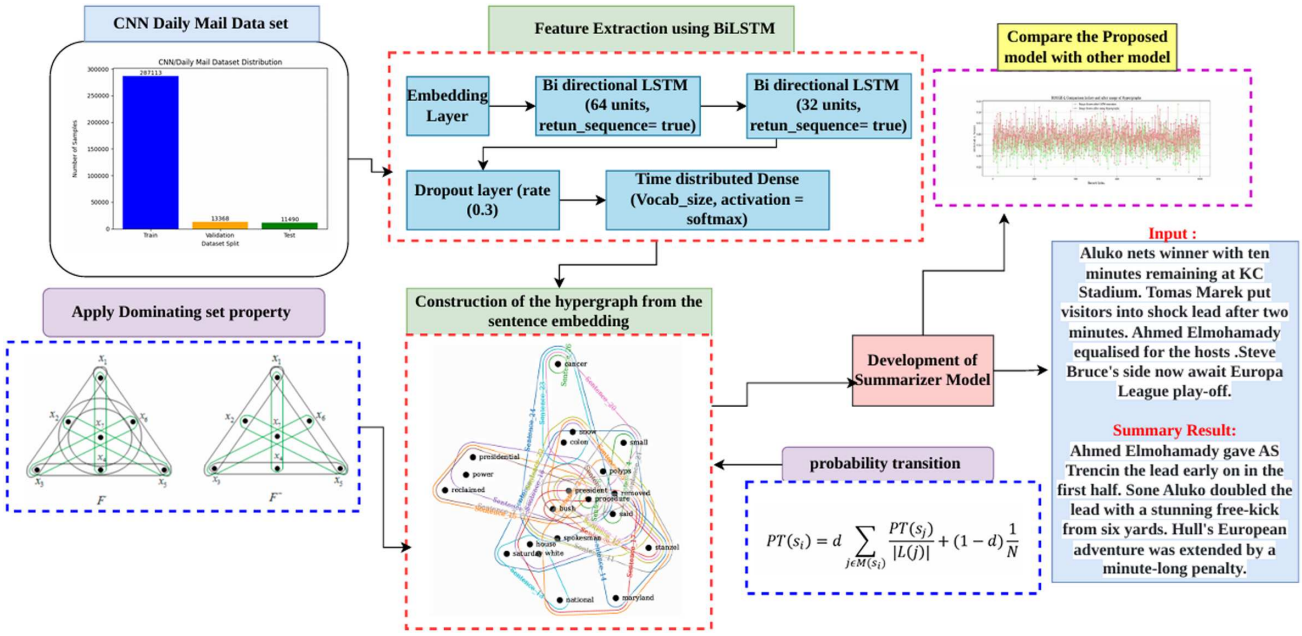
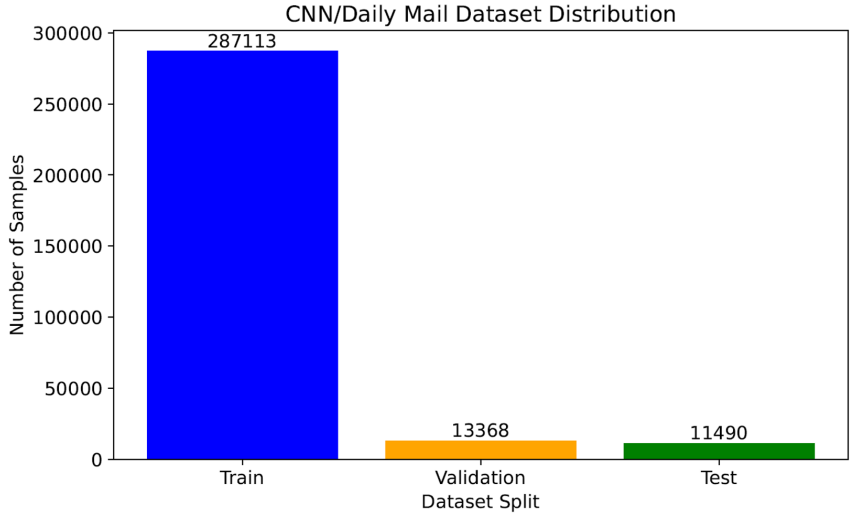


Figure 2
Data distribution (CNN/Daily Mail dataset)



embedding dimension size. The embedding layer maps a discrete sentence into dense vector representations. The resultant embedding layer is a matrix. It can be expressed by Equation (1).

$$E \epsilon R^{v \times d} \tag{1}$$

Given an input sequence of token indices $X \epsilon R^{T}$ (where T is the sequence length), the embedding layer retrieves a sentence embedding as shown in Equation (2).

$$H_0 = E[X] \, \epsilon R^{T \times d} \tag{2}$$

A BiLSTM with 64 units is used after enhancing the Embedding Layer. The BiLSTM unit at time step t performs the operations represented in Equations (3)–(7).

1. Input Gate:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_t) \tag{3}$$

2. Forget Gate:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{4}$$

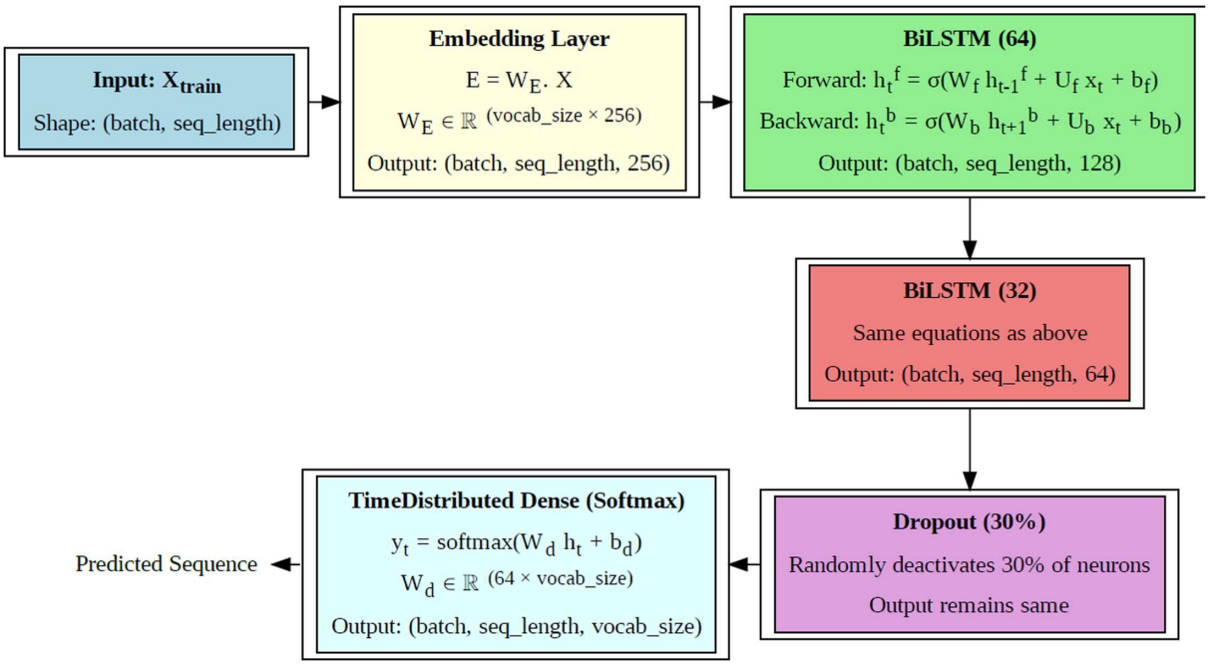
3. Cell State Update:

$$c_t = f_t \odot + c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$

4. Output Gate:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

Figure 3
Feature extraction using BiLSTM



5. Hidden State:

$$h_y = o_t \odot \tanh(c_t) \tag{7}$$

where x_t is the input vector at the time t, h_{t-1} refers to the hidden state from the previous time step, c_t denotes the cell state at the time t, W refers to the weight matrix associated with the input, U denotes the weight matrix associated with the previous hidden state, b refers to the bias terms for each gate, and the output h_t represents the feature vector for each sentence.

BiLSTM Neural Networks have two neural networks: one works forward h_t^{fwd}, and the other works backward h_t^{bwd}. The final output is represented as $H_t = \left[h_t^{fwd} h_t^{bwd}\right]$. The output dimension per direction is 64. After concatenation, the output at each time step is of size 128 or $H_1 \epsilon R^{T\times 128}$. Here, the dropout function randomly sets 30% of the activation to zero during training to prevent overfitting. $H_3 = D(H_2)$, where $D(x) = M \odot x$, M ~Bernoulli$(1-p)$. In the equation, M is a mask matrix 0^s and 1^s. After the dropout function, the output of the layer is represented with 64. This is represented as $H_3 = R^{T\times 64}$. Time distribution applies a dense layer with Softmax to each time step separately. The transformation for t is expressed in Equation 8.

$$y_t = softmax\,(w_d H_{3t} + b_d) \tag{8}$$

The Softmax function can be calculated using Equation 9.

$$P(y_t = i) = \frac{\exp(W_D[i]\,h_t + b_D[i])}{\sum_{j=1}^{vocab\_size} \exp(W_D[j]\,h_t + b_D[j]} \tag{9}$$

where, y_t represents the predicted token at the time step t, W_D and b_D are the weight matrix and bias vector of the decoder or classifier, and h_t is the hidden state at time t. The denominator sums up all possible classes (from $j = 1$ to the vocabulary size), ensuring that the output is a valid probability distribution. The output size is the vocabulary size, and the final output is $Y \in R^{T\times V}$. Finally, sparse categorical cross-entropy is used as the output, which is a probability distribution over vocab_size. The loss function is calculated using Equation 10.

$$L = -\sum_{t=1}^{T} logP(y_t|X) \tag{10}$$

where y_t is the true sentence at time t. The dense layer, S (contextualized sentence embeddings), represents a probability distribution over the vocabulary. The output sentence embedding is a 3D tensor represented in Equation 11. This is the sentence embedding result.

$$H \in R^{N\times T\times h} \tag{11}$$

Here, N is the batch size (number of sentences), T is the sequence length (number of words in a sentence), and h is the hidden state dimension of the last layer.

3.3. Hypergraph construction from extracted features

The hypergraph captures local and global sentence relationships [32, 33] and the semantic and contextual relationships, which are further used for subsequent processing to determine the dominating set for summarization. A hypergraph is constructed to identify the sentence associations using the feature vectors extracted by BiLSTM. The hypergraph is built using the sentence embedding result extracted from the neural network. The hypergraph is defined as $H = (V, E)$, where V is the set of nodes (vertices) representing unique sentence in a given preprocessed document and $E \subseteq P(V)$ is the set of hyperedges, where each hyperedge $e \in E$ is a non-empty subset. Each sentence embedding $h_i \in \mathbb{R}^h$ is considered a node in the hypergraph. Hyperedge e_k is formed using the semantic similarity between groups of sentences. The semantic similarity between two sentences, s_i and s_j, is computed using cosine similarity over the contextual feature

vectors, h_i and h_j, extracted from the BiLSTM encoder, as expressed in Equation (12).

$$Sim(S_i, S_j) = \frac{h_i.h_j}{||h_i||\,||h_j||} \tag{12}$$

If $Sim\,(s_i, s_j) > threshold$, a hyperedge is established. The threshold is determined by tuning the model's hyperparameters. Equation 13 defines the hypergraph's incidence matrix H.

$$H_{ik} = \{\frac{1\ if\ si\ belongs\ to\ ek}{0,\ otherwise} \tag{13}$$

$H_{ik} = 1$ indicates that sentence s_i is part of the hyperedge e_k.

3.4. Application of dominating set property

The dominating set property is applied to identify a subset of nodes (sentence) that overrule the hypergraph, to ensure maximum coverage of key sentences and minimize redundancy [34, 35]. Sentence Se_i is part of the dominating set D if:

$$\forall Se_j \in V,\ \exists Se_i \in D \text{ such that} (Se_i, Se_j) \in E \tag{14}$$

In Equation (14), V represents the set of all sentences in the hypergraph, D is the dominating set, representing important sentences in the document, and E denotes the hyperedges representing relationships (e.g., co-occurrence) between sentences. This selection ensures that every sentence in the hypergraph is in the dominating set or is directly associated with a sentence in the dominating set via a hyperedge. This arrangement reduces redundancy and maximizes the coverage of significant sentences. Equation 15 expresses this condition.

$$Maximize\colon \sum_{i \in D} Se_i \tag{15}$$

where Se_i is the importance of the Sentence Se_i, calculated based on its frequency and Term Frequency-Inverse Document Frequency (TF-IDF) score, or the semantic weight of the sentence is derived from BiLSTM embeddings. The dominating sentences are extracted, and their corresponding scores are computed from the hypergraph. Once the dominating set D is determined, it is used to measure the importance of a sentence using a random walk model on the hypergraph. The generation of the dominating set from the hypergraph is represented in Equation (5).

3.5. Application of PageRank algorithm

Sentences are ranked by a PageRank algorithm based on their importance in a hypergraph structure. Every sentence is regarded as a node, and the edges show the connections between sentences, like sentence co-occurrence or semantic similarity. Higher ratings indicate greater significance, and the system iteratively gives scores to sentences [36–40]. PageRank score for sentence s_i is computed as shown in Equation (16).

$$PR(s_i) = d\sum_{j \in M(s_i)} \frac{PR(s_j)}{|L(j)|} + (1-d)\frac{1}{N} \tag{16}$$

Where:

1) $PR(s_i)$ is the PageRank score of sentence s_i
2) $M(s_i)$ represents the set of sentences linking to s_i
3) L(j) denotes the number of outbound connections from s_j
4) d is the damping factor (typically set to 0.85), which controls the probability of continuing navigation through the graph
5) N is the total number of sentences in a document

This iterative calculation redistributes importance across sentences, converging to a stable ranking. The sentences with the highest scores are considered the most relevant for summarization.

3.6. Integration of dominating set property and PageRank algorithm

The PageRank algorithm and the dominating set property are applied for the selection of the most important and enlightening sentences from the text to increase the efficiency of the text summarization process. The dominating set property guarantees the coverage of all key information using a small subset of sentences that would reflect the whole text. On the other hand, the PageRank algorithm calculates the weight of each selected sentence according to its impact on the hypergraph topology. After the computation of weights, the sentences are sorted according to PageRank values in decreasing order. This is expressed by Equation 17.

$$S_{sorted} = argsort_{s \in S}(PR(s)) \tag{17}$$

Where:

1) D is the dominating set of sentences that ensures coverage
2) $PR(s)$ is the PageRank score of sentences
3) S_{sorted} is the sorted list of dominant sentences ranked by their importance

Through this ranking technique, sentences that have significant influence can be given priority to be included in the summary. This way, the important information in the text is captured without losing any coherence or creating redundancy. The combination of the two techniques ensures that the summary is brief but complete at the same time.

4. Result and Discussion

The outcomes indicate the efficacy of the proposed approach when compared to baseline approaches in numerical and qualitative terms. This part focuses on the thorough assessment of the model's efficiency through different evaluation parameters, benchmarking against baseline models, and visualizations. This dataset consists of 11,490 test samples, 13,368 validation samples, and 287,113 training samples. Figure 4 displays a sample word cloud of the records.

The log-scale graph of highlight lengths distribution by the number of highlighted words is presented in Figure 5. Since a log-scale has been chosen for the x-axis, it is much easier to track changes in length. There is also a noticeable peak on the histogram, which means that highlights are concentrated within a certain range of word numbers, specifically from 10 to 100 words. It seems that short highlights occur much more frequently than long ones. In turn, the density curve (in red color) shows smoothed data that can be compared with the histogram since the curves resemble each other.

Lengths of the article and highlights have been compared using the violin plot and box plot, presented in Figure 6(a) and (b). The articles are widely distributed and have several outliers; they are longer in comparison. The highlights, on the other hand, have little variation and are shorter in length. The articles show a wide spread, indicating variations in content durations, with highlights clustered together.

Figure 4
Word cloud for top 4 records

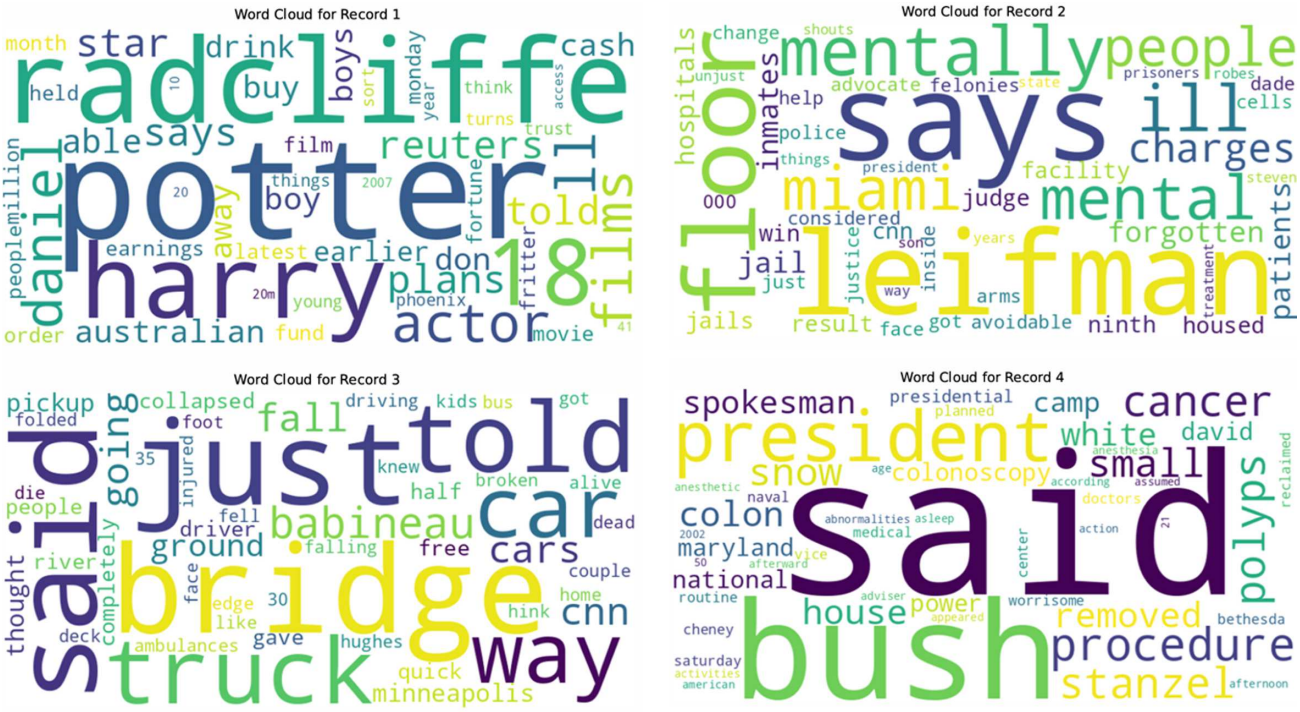
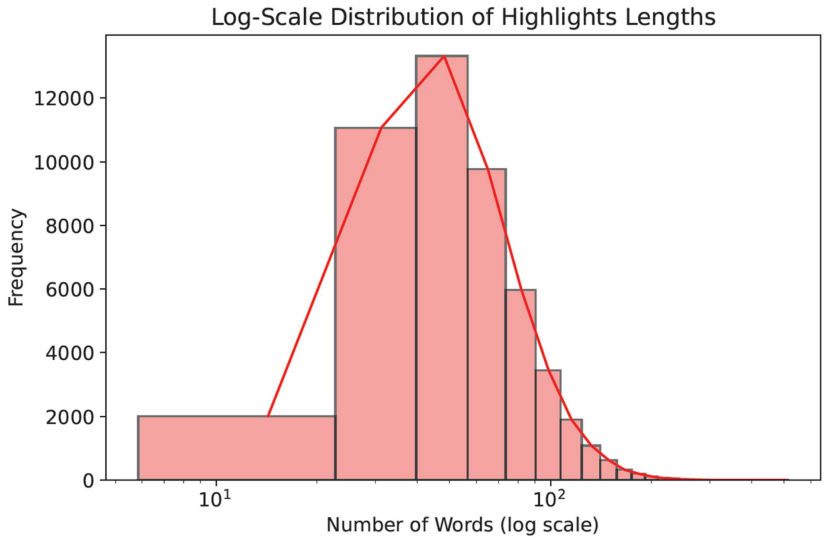


Figure 5
Word distribution among the corpus



Figures 7, 8, 9, and 10 illustrate hypergraph representation for different records. After extracting the features with BiLSTM, the hypergraph is built using the sentence embeddings. The frequently used ROUGE measure was employed to evaluate the proposed algorithm using the CNN/Daily Mail dataset. Metrics like ROUGE are often employed in natural language processing tasks to gauge the quality of summaries produced by the system compared to those manually generated. ROUGE-1, ROUGE-2, and ROUGE-L are computed using Equations 18, 19, and 20, respectively.

$$ROUGE\ 1 = \frac{Number\ of\ overlapping\ unigrams}{Total\ number\ of\ unigrams\ in\ reference\ summary} \tag{18}$$

$$ROUGE\ 2 = \frac{Number\ of\ overlapping\ bigrams}{Total\ number\ of\ bigrams\ in\ reference\ summary} \tag{19}$$

$$ROUGE\ L = \frac{LCS(generated\ summary, reference\ summary}{Total\ number\ of\ words\ in\ reference\ summary} \tag{20}$$

These metrics measure how well the generated summary communicates the key points of the reference summary. A better summary quality is reflected by a high value of the ROUGE metric score. The ROUGE scores before and after using hypergraphs are shown in Figure 11. Higher scores show better performance in

Figure 6
Comparison of article and highlight lengths

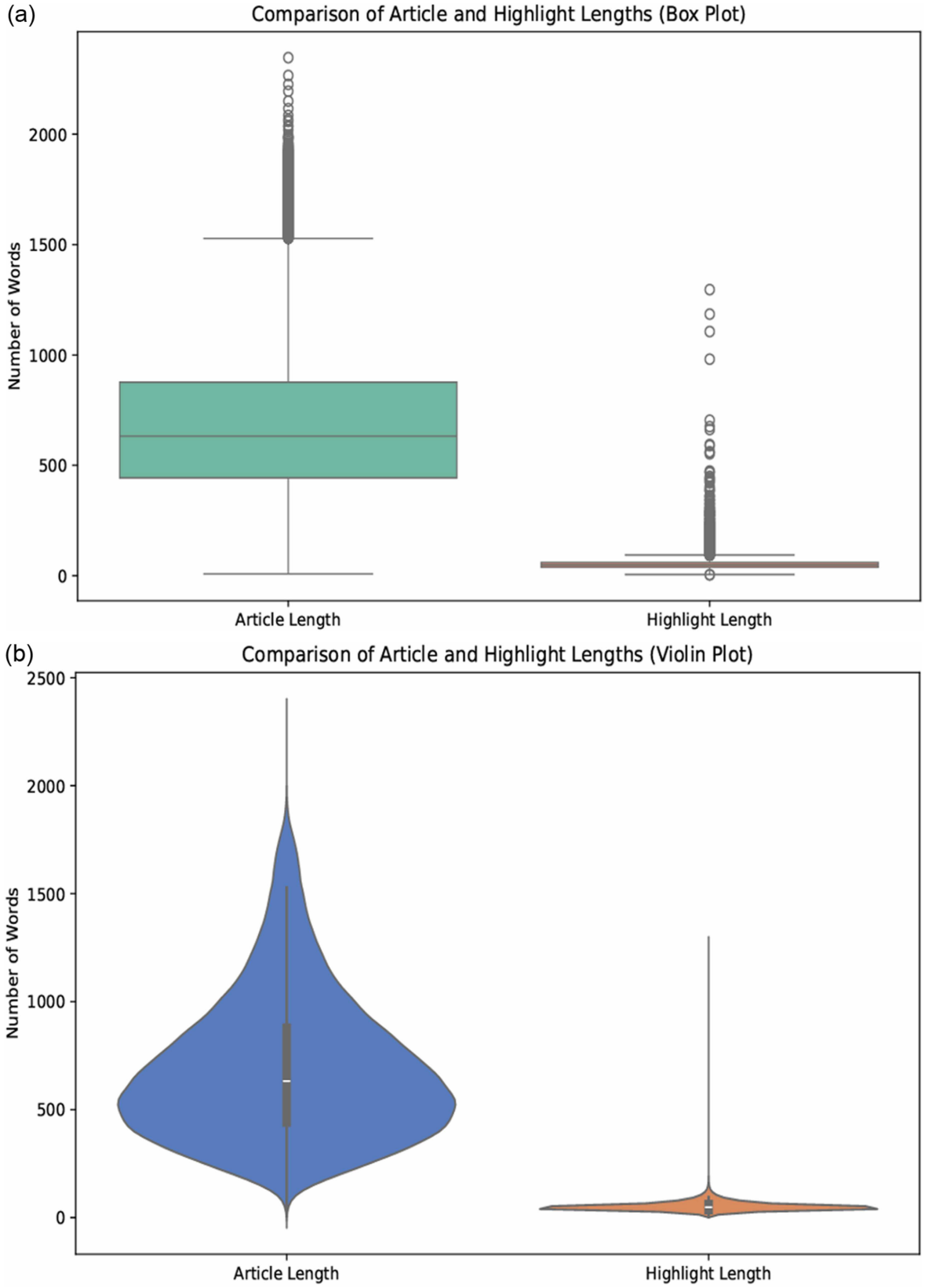


Figure 7
Hypergraph construction—Record 1

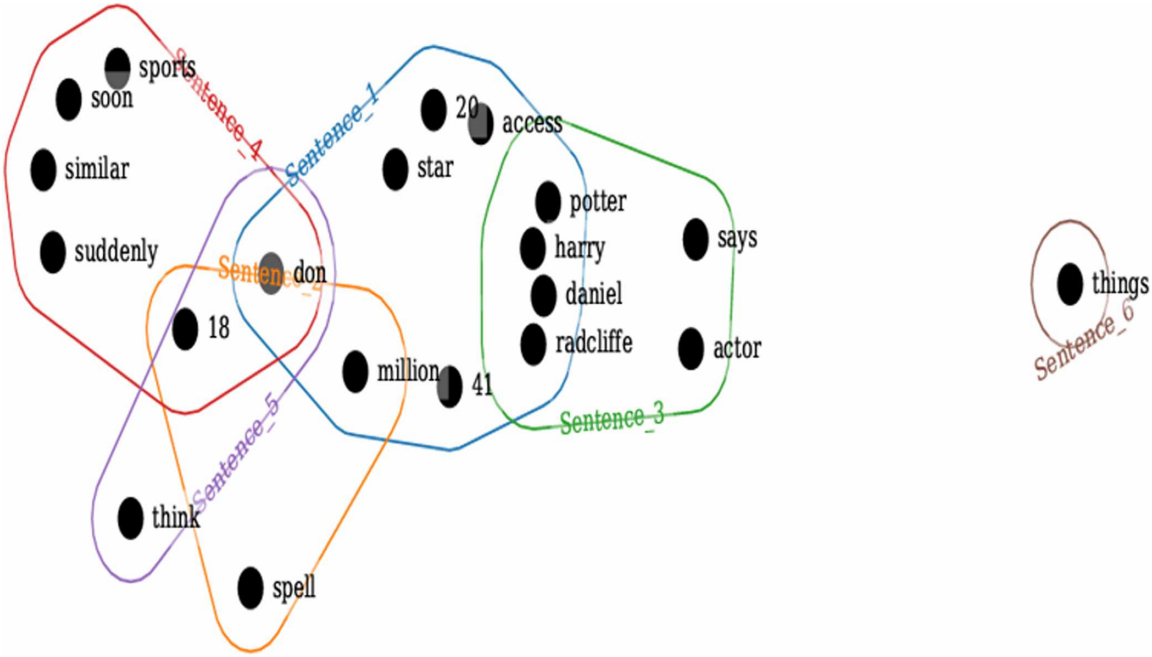


Figure 8
Hypergraph construction—Record 2

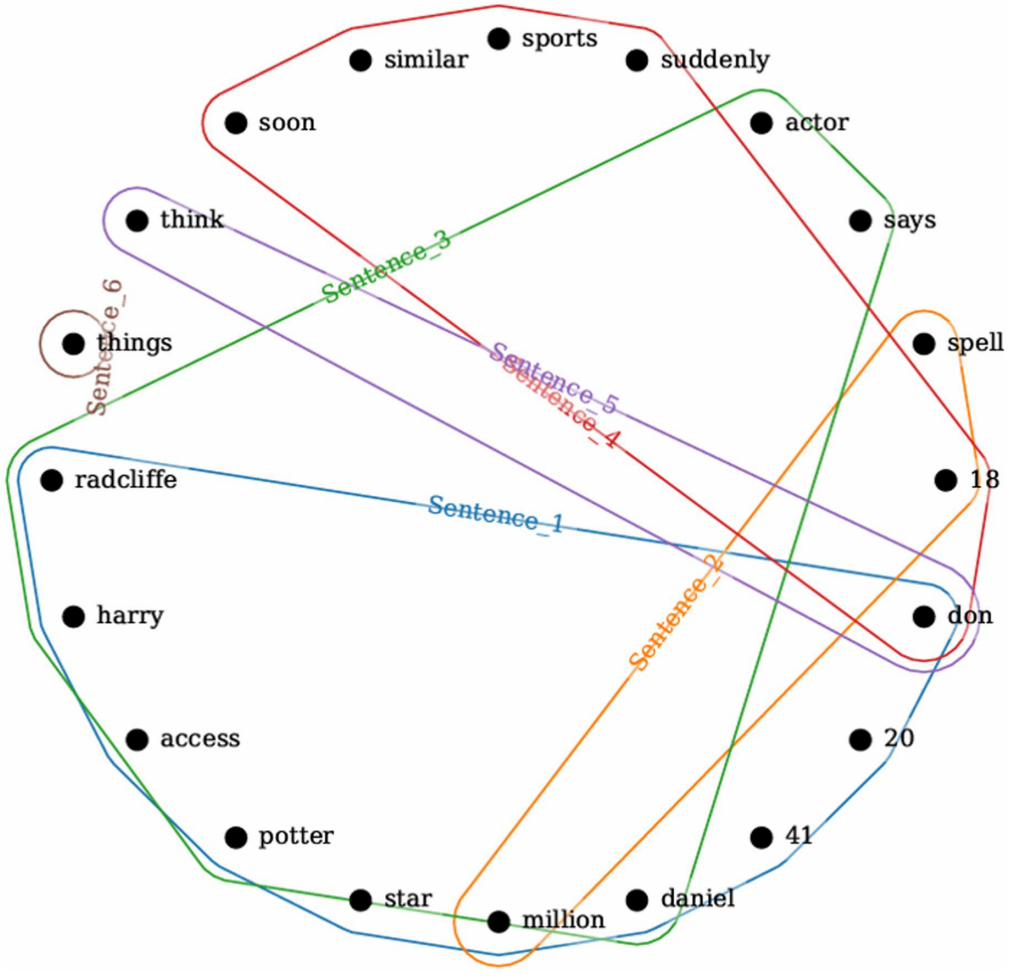


Figure 9
Hypergraph construction—Record 3

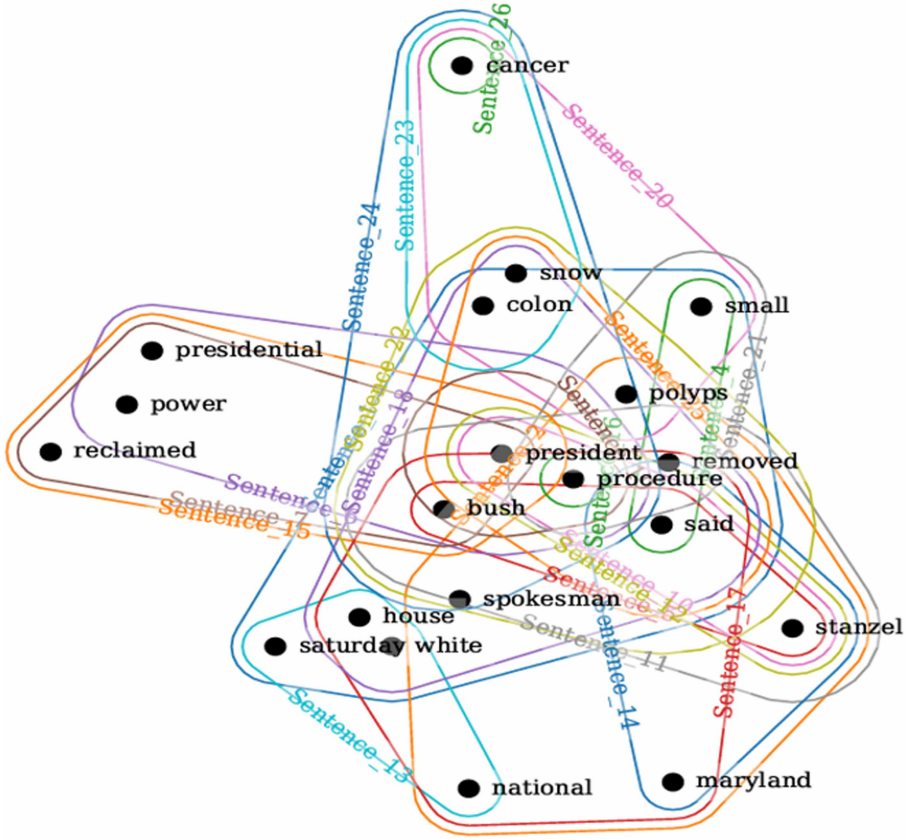


Figure 10
Hypergraph construction—Record 4

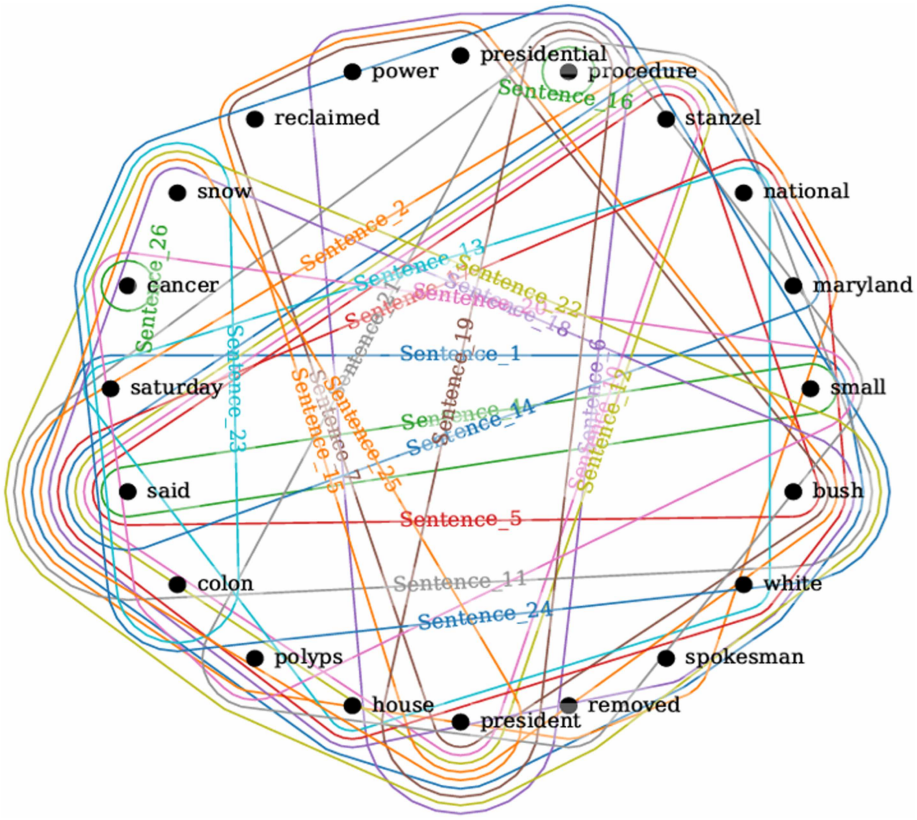
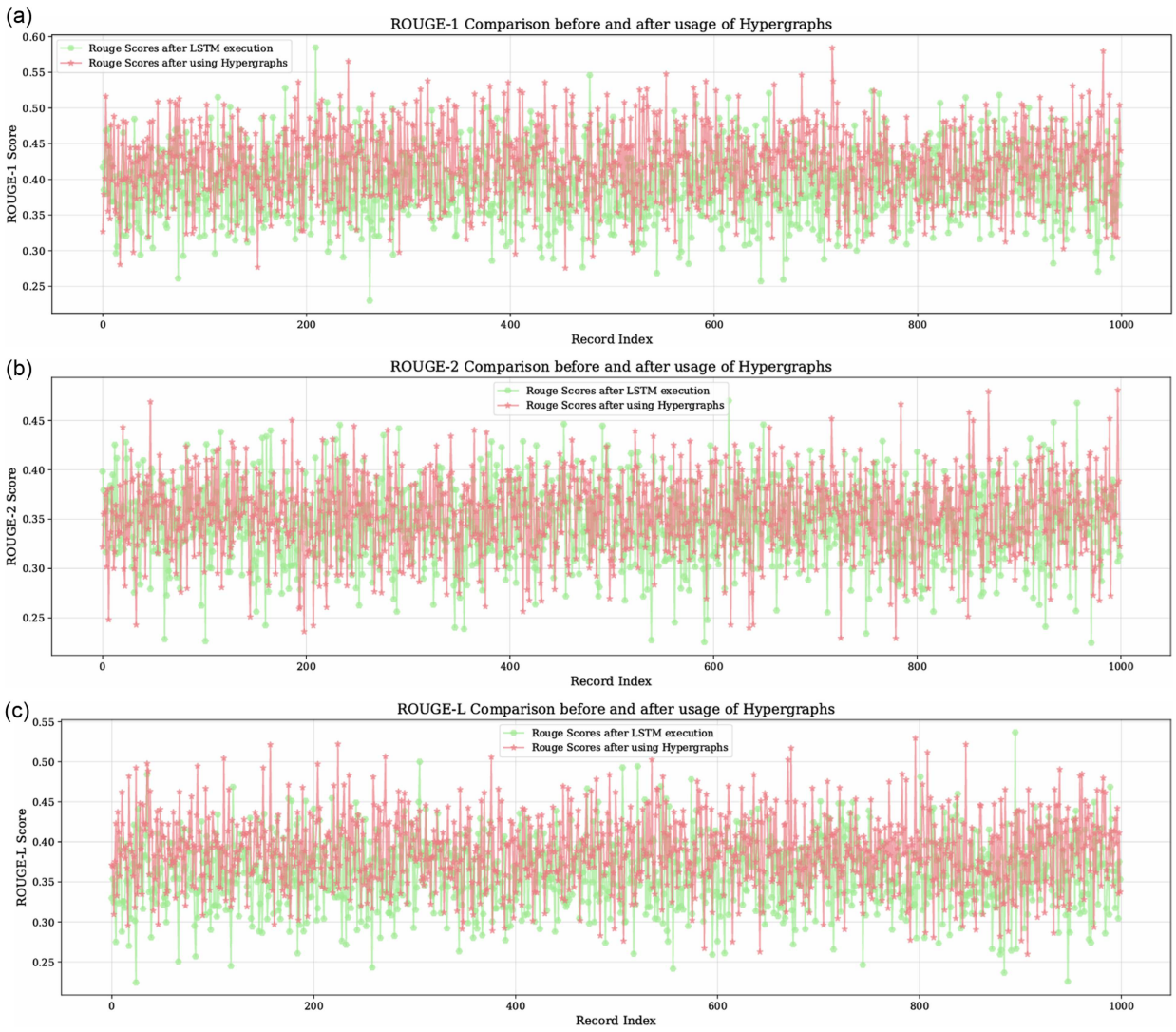


Figure 11
ROUGE-1, ROUGE-2, and ROUGE-L comparison for LSTM and hypergraph-based LSTM



the summarization task. Even though both methods display quite a wide variance, the hypergraph approach always performs better. This improvement shows that the hypergraph approach beats the LSTM in text summarization.

Table 1
Comparative analysis

Model	Rouge-1	Rouge-2	Rouge-L
TSERNN [37]	40.4	16.9	36.1
GPT-2 [38]	29.34	8.27	26.58
SummaRuNNer [40]	0.3960	0.1620	0.3530
LEAD-3 [41]	39.20	15.70	35.50
NEUSUM [41]	41.59	19.01	37.98
BiGRUExt [41]	39.42	16.13	35.11
Proposed model	0.4221	0.3563	0.3901

The tight clustering implies that the performance level remains uniform. Neither an ascending nor a descending trend exists; however, the fluctuations are indicative of changes in the degree of similarity of the text. Peaks indicate strong text overlap in some cases, while dips show weaker matches.

The comparison between the proposed model and other advanced algorithms is presented in Table 1. The proposed model produces higher ROUGE scores and shows that it is more capable than the current models including SummaRuNNer and NEUSUM in extracting both unigrams and bigrams.

5. Conclusion

In this research work, an innovative model of text summarization was developed by combining a BiLSTM network with the concept of dominating sets from hypergraphs. Semantic dependencies were accurately captured by the model, and hence redundancy was significantly minimized. The outcomes achieved showed that the framework had excellent capability as compared to other baseline models. ROUGE scores of ROUGE-1, ROUGE-2, and ROUGE-L were determined at 0.4221, 0.3563, and 0.3901, respectively. These results clearly indicate the importance of applying complex neural models and graph structures in achieving quality summarizations of textual data. Future

improvements will be realized by including extractive methods, multiple languages, and even external knowledge bases in text summarization frameworks.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail/data1.

Author Contribution Statement

Pradeepa Sampath: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Writing – review & editing, Project administration. **Shrijaa Venkatasubramanian Subashini:** Conceptualization, Methodology, Software, Data curation, Writing – original draft, Writing – review & editing. **Vimal Shanmuganathan:** Validation, Formal analysis, Investigation, Resources, Visualization, Supervision. **Seifedine Kadry:** Validation, Formal analysis, Investigation, Resources, Visualization, Supervision.

References

[1] Lian, Y., Tang, H., Xiang, M., & Dong, X. (2024). Public attitudes and sentiments toward ChatGPT in China: A text mining analysis based on social media. *Technology in Society*, *76*, 102442. https://doi.org/10.1016/j.techsoc.2023.102442

[2] Supriyono, Wibawa., P, A., Suyono, & Kurniawan, F. (2024). A survey of text summarization: Techniques, evaluation and challenges. *Natural Language Processing Journal*, *7*, 100070. https://doi.org/10.1016/j.nlp.2024.100070

[3] Liu, W., Sun, Y., Yu, B., Wang, H., Peng, Q., Hou, M., . . . , & Liu, C. (2024). Automatic text summarization method based on improved textrank algorithm and k-means clustering. *Knowledge-Based Systems*, *287*, 111447. https://doi.org/10.1016/j.knosys.2024.111447

[4] Khan, B., Usman, M., Khan, I., Khan, J., Hussain, D., & Gu, Y. H. (2025). Next-generation text summarization: A T5-LSTM FusionNet hybrid approach for psychological data. *IEEE Access*, *13*, 37557–37571. https://doi.org/10.1109/ACCESS.2025.3540590

[5] Yadav, A. K., Ranvijay, Yadav., S, R., & Maurya, A. K. (2024). Graph-based extractive text summarization based on single document. *Multimedia Tools and Applications*, *83*(7), 18987–19013. https://doi.org/10.1007/s11042-023-16199-8

[6] Shakil, H., Farooq, A., & Kalita, J. (2024). Abstractive text summarization: State of the art, challenges, and improvements. *Neurocomputing*, *603*, 128255. https://doi.org/10.1016/j.neucom.2024.128255

[7] Tank, M., & Thakkar, P. (2024). Abstractive text summarization using adversarial learning and deep neural network. *Multimedia Tools and Applications*, *83*(17), 50849–50870. https://doi.org/10.1007/s11042-023-17478-0

[8] Kiran, G. U., Gandi, R., Lavanya, M., Desale, G. B., Rao, C. U., & Reddy, B. V. (2024). Deep learning based abstractive text summarization: A survey. In *2024 Parul International Conference on Engineering and Technology*, 1–5. https://doi.org/10.1109/PICET60765.2024.10716176

[9] Radhakrishnan, P., & SenthilKumar, G. (2024). STAB: An enhanced abstractive text summarization employing stacked Bi-GRU with the attention CNN approach. *SN Computer Science*, *5*(6), 728. https://doi.org/10.1007/s42979-024-03061-3

[10] Aggarwal, D., & Sharma, A. (2025). Abstractive text summarization for legal documents using optimization-based Bi-LSTM with encoder-decoder model. *International Journal of Information Technology & Decision Making*, *24*(05), 1493–1519. https://doi.org/10.1142/S0219622025500129

[11] Gambhir, M., & Gupta, V. (2025). Improved hybrid text summarization system using deep contextualized embeddings and statistical features. *Multimedia Tools and Applications*, *84*(14), 13929–13958. https://doi.org/10.1007/s11042-024-19524-x

[12] Ding, Y., Han, S. C., Lee, J., & Hovy, E. (2026). Deep learning based visually rich document content understanding: A survey. *Artificial Intelligence Review*, *59*(4), 114. https://doi.org/10.1007/s10462-025-11477-3

[13] Teng, F., Xiao, Q., Li, X., Ye, X., & Li, Q. (2026). Self-supervised aggregation framework for text-attributed heterogeneous graphs representation. *IEEE Transactions on Knowledge and Data Engineering*, *38*(7), 4694–4706. https://doi.org/10.1109/TKDE.2026.3683100

[14] Yang, S., Yang, Y., Mi, C., Pan, Y., Wang, L., & Ma, B. (2018). Extractive summarization of documents by combining semantic content and non-structured features. In *2018 International Conference on Asian Language Processing*, 279–284. https://doi.org/10.1109/IALP.2018.8629170

[15] Huang, J. (2024). A novel extractive summarization method based on multi-path feature learning. In *2024 5th International Conference on Artificial Intelligence and Electromechanical Automation*, 448–452. https://doi.org/10.1109/AIEA62095.2024.10692601

[16] Du, K., Lu, G., & Qin, K. (2023). An extractive text summarization based on reinforcement learning. In *2023 The 6th International Conference on Software Engineering and Information Management*, 19–25. https://doi.org/10.1145/3584871.3584874

[17] Fan, J., Tian, X., Lv, C., Zhang, S., Wang, Y., & Zhang, J. (2023). Extractive social media text summarization based on MFMMR-BertSum. *Array*, *20*, 100322. https://doi.org/10.1016/j.array.2023.100322

[18] González, J. Á., Segarra, E., García-Granada, F., Sanchis, E., & Hurtado, L.-F. (2023). Attentional extractive summarization. *Applied Sciences*, *13*(3), 1458. https://doi.org/10.3390/app13031458

[19] Wang, Z., Wang, B., Dou, H., & Liu, Z. (2025). Windows deep transformer Q-networks: An extended variance reduction architecture for partially observable reinforcement learning. *Applied Intelligence*, *55*(1), 35. https://doi.org/10.1007/s10489-024-05867-3

[20] Rahman, N., & Borah, B. (2024). Query-based extractive text summarization using sense-oriented semantic relatedness measure. *Arabian Journal for Science and Engineering*, *49*(3), 3751–3792. https://doi.org/10.1007/s13369-023-07983-7

[21] Tantius, C., Shintaro, C., Soelistio, E. A., Kristanto, J., Nelson, R., & Girsang, A. S. (2024). Small2BERT for extractive text summarization. *AIP Conference Proceedings*, *2927*(1), 060016. https://doi.org/10.1063/5.0205231

[22] Wang, J.-H., & Zhuang, J.-A. (2024). Text summary generation based on data augmentation and contrastive learning. In *2024 IEEE International Conference on Big Data*, 7218–7224. https://doi.org/10.1109/BigData62323.2024.10825107

[23] Gao, Y., Li, S., Wang, P., & Wang, T. (2024). Jointly extractive and abstractive training paradigm for text summarization. In *Neural Information Processing: 30th International Conference*, 420–433. https://doi.org/10.1007/978-981-99-8181-6_32

[24] Asmitha, M., Kavitha, C. R., & Radha, D. (2024). Summarizing news: Unleashing the power of BART, GPT-2, T5, and PEGASUS models in text summarization. In *2023 4th International Conference on Intelligent Technologies*, 1–6. https://doi.org/10.1109/CONIT61985.2024.10626617

[25] Huang, J., Wu, W., Li, J., & Wang, S. (2023). Text summarization method based on gated attention graph neural network. *Sensors*, *23*(3), 1654. https://doi.org/10.3390/s23031654

[26] Adeniyi, J. K., Ajagbe, S. A., Adeniyi, A. E., Aworinde, H. O., Falola, P. B., & Adigun, M. O. (2024). EASESUM: An online abstractive and extractive text summarizer using deep learning technique. *IAES International Journal of Artificial Intelligence*, *13*(2), 1888. https://doi.org/10.11591/ijai.v13.i2.pp1888-1899

[27] Ajagbe, S. A., Oladosu, J. B., Falohun, A. S., & Olayiwola, A. A. (2026). Thresholding in convolutional neural network model for yorùbá speech-to-text conversion. *FUOYE Journal of Engineering and Technology*, *10*(3), 456–463. https://doi.org/10.4314/fuoyejet.v10i3.10

[28] Vo, S.-N., Vo, T.-T., & Le, B. (2024). Interpretable extractive text summarization with meta-learning and BI-LSTM: A study of meta learning and explainability techniques. *Expert Systems with Applications*, *245*, 123045. https://doi.org/10.1016/j.eswa.2023.123045

[29] Babu, G. L. A., & Badugu, S. (2024). Multi-document hybrid text summarization with bi-LSTM RNN for Telugu language. *Sādhanā*, *49*(2), 155. https://doi.org/10.1007/s12046-024-02499-8

[30] Wazery, Y. M., Saleh, M. E., Alharbi, A., & Ali, A. A. (2022). Abstractive Arabic text summarization based on deep learning. *Computational Intelligence and Neuroscience*, *2022*(1), 1566890. https://doi.org/10.1155/2022/1566890

[31] Huang, J., Pu, Y., Zhou, D., Cao, J., Gu, J., Zhao, Z., & Xu, D. (2024). Dynamic hypergraph convolutional network for multimodal sentiment analysis. *Neurocomputing*, *565*, 126992. https://doi.org/10.1016/j.neucom.2023.126992

[32] Pradeepa, S., Jomy, E., Vimal, S., Hassan, M. M., Dhiman, G., Karim, A., & Kang, D. (2024). HGATT_LR: Transforming review text classification with hypergraphs attention layer and logistic regression. *Scientific Reports*, *14*(1), 19614. https://doi.org/10.1038/s41598-024-70565-6

[33] Kobayashi, Y., Kurita, K., Mann, K., Matsui, Y., & Ono, H. (2025). Enumerating minimal vertex covers and dominating sets with capacity and/or connectivity constraints. *Algorithms*, *18*(2), 112. https://doi.org/10.3390/a18020112

[34] Bonnet, É., Geniet, C., Kim, E. J., Thomassé, S., & Watrigant, R. (2024). Twin-width III: Max independent set, min dominating set, and coloring. *SIAM Journal on Computing*, *53*(5), 1602–1640. https://doi.org/10.1137/21M142188X

[35] Li, D., Cheng, C., Zhao, X., & Li, Z. (2025). Port node importance and risk assessment in shipping networks based on an improved PageRank approach and reliability analysis. In *2024 International Conference on Smart Transportation Interdisciplinary Studies*, 01947841. https://doi.org/10.4271/2025-01-7183

[36] Charikhi, M. (2024). Association of the PageRank algorithm with similarity-based methods for link prediction in complex networks. *Physica A: Statistical Mechanics and its Applications*, *637*, 129552. https://doi.org/10.1016/j.physa.2024.129552

[37] Hosni, A. I. E., Baira, I., Djamaa, B., Sidhoum, Hamouda., Merini, A., H, & Amrani, M. C. (2026). A line graph-based metric learning framework for robust link prediction in complex networks. *The Journal of Supercomputing*, *82*(9), 512. https://doi.org/10.1007/s11227-026-08645-9

[38] Zhang, Y., Liao, J., Tang, J., Xiao, W., & Wang, Y. (2018). Extractive document summarization based on hierarchical GRU. In *2018 International Conference on Robots and Intelligent Systems*, 341–346. https://doi.org/10.1109/ICRIS.2018.00092

[39] Suleiman, D., & Awajan, A. A. (2019). Deep learning based extractive text summarization: Approaches, datasets and evaluation measures. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security*, 204–210. https://doi.org/10.1109/SNAMS.2019.8931813

[40] Zhang, C., Wang, J., Qi, H., Yan, C., & Jiang, C. (2021). Multi-view metrics enhanced heterogeneous graph neural network for extractive summarization. In *2021 China Automation Congress*, 3180–3185. https://doi.org/10.1109/CAC53003.2021.9727996

How to Cite: Sampath, P., Subashini, S. V., Shanmuganathan, V., & Kadry, S. (2026). Efficient Extractive Text Summarization Using BiLSTM, Hypergraph, and Dominating Set Property. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62029367