Extractive Text Summarization-Based Framework for Sindhi Language

Aqsa Memon, Zainab Memon, Akhtar Hussain Jalbani
Department of Computer Science, Quaid-e-Awam University of Engineering Science & Technology Nawabshah, Sindh
*Correspondence: memonaqsa695@gmail.com , zainabmemon994@gmail.com

Citation | Memon. A, Memon. Z, Jalbani. A. H, "Extractive Text Summarization-Based Framework for Sindhi Language", IJIST, Vol. 07 Special Issue. pp 147-155, May 2025
Received| April 18, 2025 **Revised|** May 15, 2025 **Accepted|** May 17, 2025 **Published|** May 19, 2025.

This paper presents an extractive text summarization method specially designed for Sindhi, a culturally rich but low-resource Indo-Aryan language spoken widely in Pakistan. The study focuses on selecting the most relevant sentences from Sindhi texts, employing Natural Language Processing (NLP) techniques to generate concise summaries.

The proposed system incorporates essential preprocessing steps, including text cleaning, tokenization, and stemming/lemmatization. For future extraction, it utilizes TF-IDF and sentence embeddings. After scoring the sentences, the most significant ones are selected to form the final summary.

To evaluate the system's performance in five test paragraphs, several metrics are used, including F1 score, precision, recall, cosine similarity, normalization level distance, and accuracy. The system demonstrates reliable and accurate summarization, and consistency achieving high precision (1.0), strong F1 score (0.89-0.92), a low a low normalized error (0.04), and an overall accuracy of 0.86. Graphic analysis further confirms the model's coherence, semantic retention, and low error rates.

By leveraging NLP for information summarization, this study contributes to preserving and promoting the Sindhi language—potential applications including digital accessibility, education, and content curation. Future research aims to enhance contextual understanding by exploring transformer-based models like BERT and extending the approach to abstraction summarization.

Keywords: Sindhi Language, Extractive Summarization, Natural Language Processing (NLP), Sentence Selection, TF-IDF, Sentence Embeddings

Introduction:

Artificial intelligence (AI) has emerged as a transformative technology, revolutionizing industries and reshaping our world. At its core, AI aims to equip machines with human-like intelligence, enabling them to learn, reason, and make decisions. This powerful technology holds the potential to solve complex problems, automate tasks, and unlock new opportunities. AI's broad applications span healthcare, finance, autonomous vehicles, customer service, and education.

One significant subfield of AI is Natural Language Processing (NLP) which seeks to bridge the gap between human language and machine understanding. NLP involves analyzing interpreting, and generating human language using advanced algorithms and statistical models. This approach enables machines to comprehend and respond to human language with nuance and contextual awareness. Application of NLP includes information retrieval, sentiment analysis, machine translation, and dialogue systems, making it essential for language recognition projects involving languages like English, Chinese, Urdu, and more.

The Sindhi language, with its deep historical roots, is the second-most spoken language in Pakistan and one of the oldest Indo-Aryan languages, closely linked to the region of Sindh. It has a rich literary tradition, encompassing poetry, prose, short stories, and other genres. For Pakistani students, working on Sindhi language projects offers a meaningful way to support the preservation and promotion of their regional language.

Extractive text summarization is a technique that identifies and selects the most important sentences from a document to produce a concise summary. While extensively researched for English and other major languages, applying this method to low-resource languages like Sindhi presents unique challenges. The primary obstacle is the limited availability of high-quality annotated Sindhi language data, essential for training robust machine learning models. To overcome this, researchers employ methods such as data augmentation, transfer learning, and unsupervised learning.

In the context of Sindhi, extractive summarization offers several benefits, including preserving cultural heritage, enhancing information accessibility, supporting language, and enabling text mining and analysis. However, implementing extractive summarization for Sindhi requires careful attention to language-specific aspects like complex sentence structures, rich morphological features, and the nuances of Sindhi grammar. Addressing these challenges through advances in NLP techniques can help develop effective summarization models that aid in preserving understanding, and sharing Sindhi language and culture.

Related Work:

Abu Nada & Abdullah M [1]**:**

Proposed an Arabic text summarizer utilizing AraBERT with clustering and Natural Language Understanding (NLU). While effective, it encounters challenges with sentence boundaries and handling long texts, highlighting areas for improvement.

Ferreira, Rafael [2]**:**

Focused on sentence scoring techniques, emphasizing methods like word frequency, TF-IDF, lexical similarity, and sentence length as effective strategies. The TextRank algorithm also demonstrated potential, underscoring the significance of sentence scoring in summarization.

Miller, Derek [3]**:**

Introduced a Python-based "lecture summarization service" that employs BERT and KMeans clustering. This service allows for customizable summary lengths and shows improved accuracy compared to traditional approaches.

Sinha, Aakash, et al.[4]**:**

Developed a data-driven summarization method using a feedforward neural network. Its ability to automatically extract features and scale to longer documents is noteworthy, though controlling the summary length remains a challenge.

Xu, Jiacheng [5]**:**

Presented DISCOBERT, a discourse-aware model leveraging Graph Convolutional Networks (GCNs) and Rhetorical Structure Theory (RST) trees. This model reduces redundancy and enhances coherence, establishing new benchmarks in extractive summarization.

Mutlu, Begum, et al.[6]**:**

Focused on dataset creation and summarization techniques by developing a model that combines syntactic and semantic features using LSTM-based networks. The model achieved superior performance by effective summaries, outperforming baseline methods and reducing redundancy.

Ruan, Qian et al.[7]**:**

Introduced HiStruct+, a transformer-based model that encodes hierarchical structure information. It delivers concise and informative summaries, outperforming baseline methods and reducing redundancy.

Gambhir, Mahak et al.[8]**:**

Conducted a comprehensive review of extractive summarization techniques from the past decade, analyzing their strengths, limitations, and applications. They highlight the need for advanced feature engineering and improved evaluation metrics in future research.

Fang, Changjian [9]**:**

Proposed CoRank, a graph-based co-ranking model that improves sentence scoring through iterative refinement of word weights. The model ensures both theoretical convergence and computational efficiency.

Fatima Zainab [10]**:**

Developed a heuristic-based summarization model that strikes a balance between compression and content retention. It outperforms existing methods while maintaining similar topic coverage to advanced models like LDA.

Objectives and Discussion section:

1. Collecting Sindhi Textual Data

Sindhi text is gathered from online newspapers, books, blogs, and official documents. Due to limited resources in Sindhi, data collection requires careful curation and possibly manual effort.

2. Preprocessing and Feature Extraction

The text is cleaned by removing noise, normalizing the script, and tokenizing sentences. Important features like TF-IDF scores, sentence position, and word embeddings are extracted to help identify key content.

3. Applying Machine Learning for Summarization

Extractive methods like TextRank or supervised models (e.g., SVM) are used to select important sentences. If resources permit, multilingual models like mBERT or mT5 can be fine-tuned for abstractive summarization.

4. Evaluation

Summaries are evaluated using ROUGE scores, Levenshtein distance, and F1-score. Human evaluation is also important for checking fluency and relevance, especially in low-resource languages like Sindhi.

Proposed Methodology:

Figure 1 illustrates the Sindhi text summarization process, which begins with data collection. Next, preprocessing steps are applied, including text cleaning, tokenization, and stemming/lemmatization. Feature extraction follows, using TF-IDF and sentence embeddings

to assess sentence importance. The proposals are then categorized based on these features, and the most significant ones are selected to form a concise summary. Finally, the generated summary is evaluated by comparing it with human-created summaries to ensure quality and accuracy.

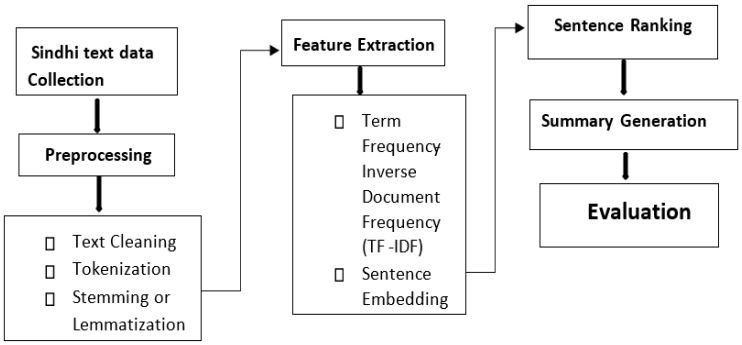

Figure 1. This flowchart depicts the process of Sindhi text summarization, including data collection, preprocessing, feature extraction, sentence ranking, summary generation, and evaluation

Results and Discussion:

Summarization Evaluation Results: The table presents metrics for a five-paragraph extractive summarization system, which consistently demonstrates high performance. It shows strong cosine similarity (0.89-0.95), robust F1 scores (0.89-0.92), and perfect precision (1.0). With a low normalized Levenshtien error of 0.04 and an overall accuracy of 0.86, the system produces clear and precise summaries with minimal text variance.

Table 1. Illustrates three key metrics used to evaluate the effectiveness of a summarization model across five distinct paragraphs: F1 score, Precision, and Recall.

Paragraph	Precision	Recall	F1 Score	Error Distance	Cosine Similarity	Accuracy
1	1.0	0.84	0.92	0.23	0.95	0.86
2	1.0	0.84	0.92	0.22	0.95	0.87
3	1.0	0.84	0.91	0.17	0.89	0.86
4	1.0	0.84	0.91	0.18	0.89	0.86
5	1.0	0.80	0.89	0.21	0.93	0.86

✅ Overall Accuracy (Mean Accuracy Score): 0.86
⚠️ Overall Error (Mean Normalized Levenshtein Distance): 0.04

Recall measures how many relevant sentences the model correctly selected, while Precision indicates the proportion of selected sentences that are genuinely relevant.

The F1 score, as the harmonic mean of precision and recall, provides a balanced measure of the model's performance.

The graph visually demonstrates the model's consistency across various paragraphs. While the model's slightly lower recall suggests it may miss some relevant content, its consistently high precision indicates strong accuracy in selecting pertinent information. Overall, the graph serves as a valuable performance analysis tool, assessing the precision, coverage, and balance of the summarization method.

The y-axis displays the normalized edit distance (error), while the x-axis shows the paragraph numbers (1-5).

A lower value indicates a more accurate summary, as fewer changes are needed to match the reference.

The graph clearly shows that the error rate remains consistently low across all paragraphs, with paragraph 3 exhibiting the lowest error. As a character-level assessment tool, this graph effectively demonstrates how closely the generated summaries align with the original ones in terms of precise wording.

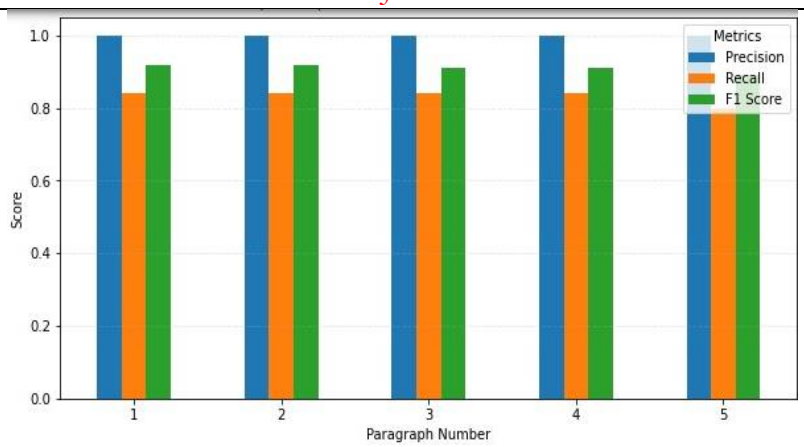

Figure 2. This graph for Precision, Recall, and F1 Score for Extractive Summarization

Figure 3 presents the Normalized Levenshtien Distance, a metric used to evaluate the error rate of a summarization model. The measure calculates the textual difference between the generated summary and the reference summary by counting the number of single-character changes required to transform one into the other, normalized by length.

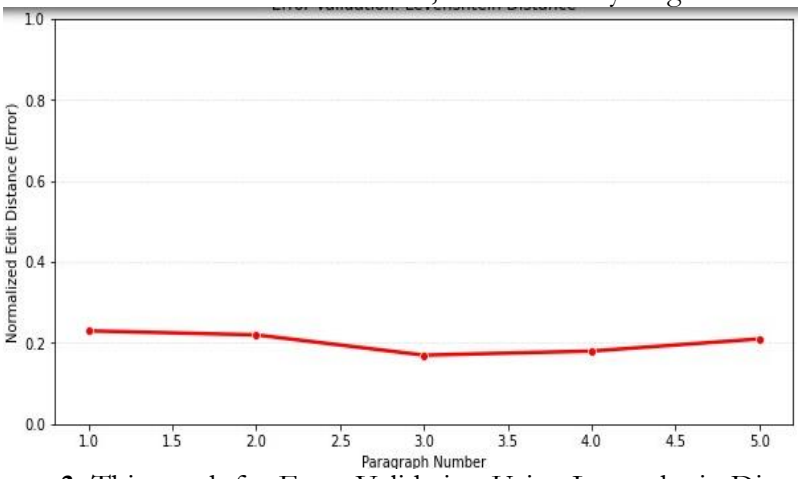

Figure 3. This graph for Error Validation Using Levenshtein Distance

Figure 4 displays the cosine similarity score between the generated and reference summaries, used to evaluate the semantic accuracy of a summarization model.

The y-axis shows cosine similarity values ranging from 0 to 1, where a value closer to 1 indicates a higher degree of semantic similarity.

The x-axis presents different paragraphs:

The graph demonstrates that the model consistently maintains strong semantic similarity across all five paragraphs, with scores consistently above 0.89. Although there is a slight decrease in similarity for paragraph 3, the overall trend indicates that the generated summaries effectively capture the main ideas of the source texts.

This graph is crucial for analyzing how well the summarizer preserves the semantic content, even when wording or structure changes.

The graph reveals that all five paragraphs consistently exhibit high accuracy scores. The accuracy starts at approximately 0.86 for the first paragraph, slightly increases to around 0.87 for the second, and then stabilizes at about 0.86 for the third, fourth, and fifth paragraphs. The minimal fluctuations indicate the model's consistently high performance in accurately summarizing the content.

Overall, the trend demonstrates that the summarization model performance is reliable and consistent across various input texts.

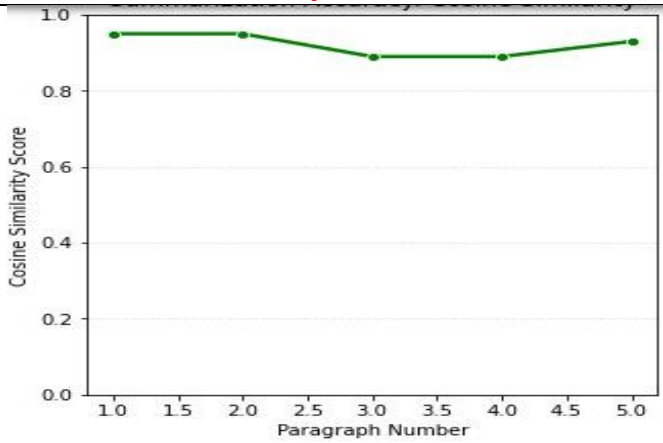

Figure 4. This graph for Summarization Accuracy Using Cosine Similarity

Figure 5 presents a line graph depicting the overall accuracy score of the summarization task based on five distinct paragraphs.

The y-axis represents the accuracy score, ranging from 0 to 1.

The x-axis shows the paragraph numbers, from 1 to 5.

The plotted line indicates the accuracy achieved for each paragraph.

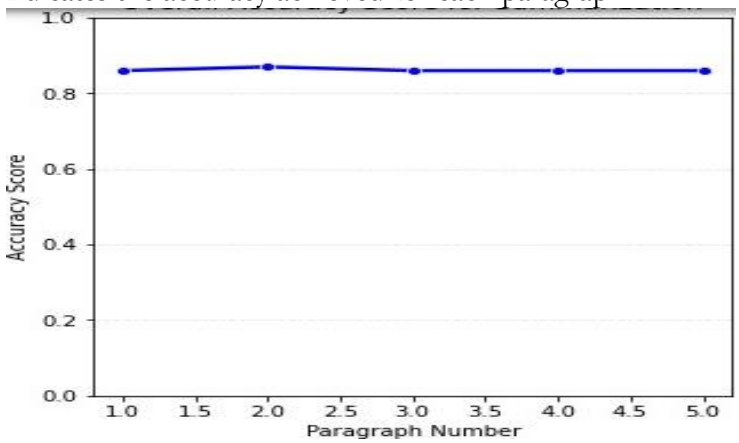

Figure 5. This graph for the Overall Accuracy Score for Summarization Across Paragraphs

Figure 6 presents a graph illustrating the simulated training error of the model evaluated across five distinct paragraphs. The graph features two error metrics:

Levenshtien Distance error (represented by the red line)

"1- Accuracy" error (represented by orange line)

The Levenshtien Distance error shows some variation between paragraphs. It starts around 0.23, decreases slightly for the second paragraph, drops significantly to approximately 0.17 for the third paragraph, and then gradually increases again for all remaining paragraphs.

In contrast, the trend of the "1 – Accuracy" error is more consistent. It begins around 0.14, decreases slightly for the second paragraph, remains relatively stable for the third and fourth paragraphs, and shows a minor increase for the fifth paragraph.

The pattern indicates that while the character-level discrepancies (as captured by the Levenshtien Distance) varied more noticeably between paragraphs, the model's overall accuracy remained relatively stable throughout training. Despite fluctuations in edit distance, the relatively consistent "1 – Accuracy" error suggests the model maintained steady performance in terms of accurate predictions.

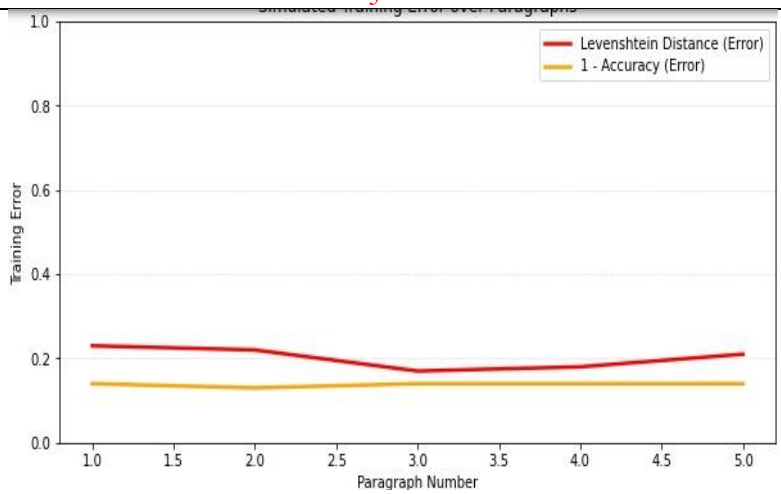

Figure 6. This graph for Simulated Training Error Over Paragraphs

Figure 7 presents a bar graph showing the "Average Accuracy" across five folds of cross-validation procedure.

The x-axis represents the "Folds", numbered 1 through 5, displaying the results from the five-fold across-validation process.

The y-axis shows the "Average Accuracy", ranging from 0.0 to 1.0.

Each bar in the graph represents the average accuracy achieved in one of the five folds. The graph demonstrates consistently high accuracy across all folds. The first fold shows an average accuracy of approximately 0.87, while the average accuracy for folds 2 through 5 is slightly lower but still comparable, around 0.86.

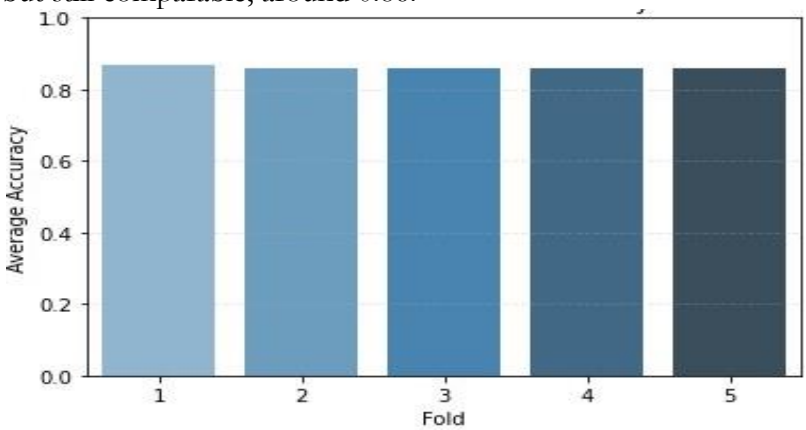

Figure7. This graph for K- Fold CrossValidation Accuracy

Conclusion:

This study introduces a reliable and effective extractive text summarization method specifically designed for Sindhi, a historically significant but computationally underrepresented language. The proposed system integrates structured preprocessing steps such as text cleaning, tokenization, and stemming/lemmatization, alongside traditional Natural Language Processing (NLP) techniques like Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction. These techniques help address the challenges associated with low-resource languages, such as the lack of annotated corpora, complex sentence structures, and diverse morphological patterns.

The algorithm generates concise and meaningful summaries by ranking sentences based on their relevance and selecting the top-ranked ones. Experimental results show the system's reliability across various evaluation metrics. Notably, it achieved a perfect precision score of 1.0 for all five evaluated paragraphs, indicating that the selected sentences were always relevant. Cosine similarity scores ranged from 0.89 to 0.95, and F1 scores varied from 0.89 to

0.92, reflecting strong semantic similarity between the generated and human-written summaries. The system also attained an overall accuracy of 0.86 and maintained a low Mean Normalized Levenshtein Error of 0.04, showing minimal textual variation. These results validate the system's ability to generate summaries that closely match human interpretations in both structure and meaning.

While the performance is promising, the study identifies several areas for improvement and suggests a path forward. One key area is expanding from extractive to abstractive summarization. Extractive methods excel at sentence selection but often struggle with information reorganization, a challenge that abstractive models, using transformers or sequence-to-sequence learning, could address more effectively. Future research will explore incorporating advanced deep learning models such as mBERT, BERT, and other transformer-based architectures adapted for low-resource languages like Sindhi. These models can significantly enhance the summarizer's ability to generate and understand semantics.

Another important direction is the development of larger, better-annotated datasets for Sindhi, as the lack of such resources currently limits the performance and generalizability of machine learning models. Collaborations with regional language boards, linguistic experts, and academic institutions could facilitate the creation of these corpora. Furthermore, techniques like cross-lingual learning, data augmentation, and transfer learning can improve model performance while mitigating data limitations.

To enhance the coherence and contextual depth of the summaries, future work could incorporate discourse-aware summarization techniques that account for sentence relationships, such as those based on Rhetorical Structure Theory (RST) or co-reference resolution. Exploring hierarchical models that consider features at the paragraph or page level could also yield more insightful results.

Finally, practical applications of this system in fields like media, education, governance, and digital content curation could provide significant benefits. For example, systems that condense lengthy Sindhi texts could be valuable for teachers and students, while news organizations and local government websites could use automated summaries to make information more accessible to Sindhi-speaking communities.

By addressing these areas and continuing to innovate, this study lays the groundwork for the future development of NLP tools for Sindhi. In doing so, we can contribute to the digital preservation of Sindhi and support the broader goal of empowering underrepresented languages in the age of artificial intelligence.

References:

[1] S. S. A.-N. Nada, Abdullah M. Abu, Alajrami, Eman, Al-Saqqa, Ahemd A., "Arabic Text Summarization Using AraBERT Model Using Extractive Text Summarization Approach," *Int. J. Acad. Inf. Syst. Res.*, vol. 4, no. 8, pp. 6–9, 2020, [Online]. Available: http://ijeais.org/wp-content/uploads/2020/8/IJAISR200802.pdf

[2] L. F. Rafael Ferreira, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Fred Freitas, George D.C. Cavalcanti, Rinaldo Lima, Steven J. Simske, "Assessing sentence scoring techniques for extractive text summarization," *Expert Syst. Appl.*, vol. 40, no. 14, pp. 5755–5764, 2013, doi: https://doi.org/10.1016/j.eswa.2013.04.023.

[3] Derek Miller, "Leveraging BERT for extractive text summarization on lectures," *arXiv:1906.04165*, 2019, doi: https://doi.org/10.48550/arXiv.1906.04165.

[4] A. G. Aakash Sinha, Abhishek Yadav, "Extractive text summarization using neural networks," *arXiv:1802.10137*, 2018, doi: https://doi.org/10.48550/arXiv.1802.10137.

[5] J. L. Jiacheng Xu, Zhe Gan, Yu Cheng, "Discourse-Aware Neural Extractive Text Summarization," *Assoc. Comput. Linguist.*, pp. 5021–5031, 2020, doi: 10.18653/v1/2020.acl-main.451.

[6] M. A. A. Begum Mutlu, Ebru A. Sezer, "Candidate sentence selection for extractive text

summarization," *Inf. Process. Manag.*, vol. 57, no. 6, p. 102359, 2020, doi: https://doi.org/10.1016/j.ipm.2020.102359.

[7] G. R. Qian Ruan, Malte Ostendorff, "HiStruct+: Improving Extractive Text Summarization with Hierarchical Structure Information," *Assoc. Comput. Linguist.*, pp. 1292–1308, 2022, doi: 10.18653/v1/2022.findings-acl.102.

[8] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 1–66, Jan. 2017, doi: 10.1007/S10462-016-9475-9/METRICS.

[9] Z. W. Changjian Fang, Dejun Mu, Zhenghong Deng, "Word-sentence co-ranking for automatic extractive text summarization," *Expert Syst. Appl.*, vol. 72, 2017, doi: 189-195.

[10] L. F. N. Waseemullah, Zainab Fatima, Shehnila Zardari, Muhammad Fahim, Maria Andleeb Siddiqui, Ag. Asri Ag. Ibrahim, Kashif Nisar, "A Novel Approach for Semantic Extractive Text Summarization," *Appl. Sci.*, vol. 12, no. 9, p. 4479, 2022, doi: https://doi.org/10.3390/app12094479.