Systematic TextRank Optimization in Extractive Summarization

Morris Zieve, Anthony Gregor, Frederik Juul Stokbaek,
Hunter Lewis, **Ellis Marie Mendoza**, and **Benyamin Ahmadnia**
Department of Computer Engineering and Computer Science
California State University, Long Beach, United States
morris.zieve01@student.csulb.edu, anthony.gregor01@student.csulb.edu,
frederikjuul.stokbaek01@student.csulb.edu, hunter.lewis01@student.csulb.edu,
ellismarie.mendoza01@student.csulb.edu, benyamin.ahmadnia@csulb.edu

Abstract

With the ever-growing amount of textual data, extractive summarization has become increasingly crucial for efficiently processing information. The TextRank algorithm, a popular unsupervised method, offers excellent potential for this task. In this paper, we aim to optimize the performance of TextRank by systematically exploring and verifying the best preprocessing and fine-tuning techniques. We extensively evaluate text preprocessing methods, such as tokenization, stemming, and stopword removal, to identify the most effective combination with TextRank. Additionally, we examine fine-tuning strategies, including parameter optimization and incorporation of domain-specific knowledge, to achieve superior summarization quality.

1 Introduction

In the modern era, the sheer volume of data generated daily poses a significant challenge for decision-makers to stay informed about the latest trends and developments. Text summarization addresses this issue by extracting only the most salient information from a text. This study investigates the effectiveness of TextRank, an extractive text summarization algorithm, compared to other common approaches, such as abstractive and hybrid summarizations.

Automatic text summarization can be classified based on the input size, algorithm, content, domain, language, type, and approach (Bounab et al., 2019). One approach is extractive summarization, which selects essential sentences from the input document(s) and concatenates them to form the summary. Another approach is abstractive summarization, which creates an intermediate representation of the input document(s) and generates a summary. Lastly, hybrid summarization combines extractive and abstractive approaches (Ansary, 2021).

Extractive Text Summarization is a widely-used approach in Natural Language Processing (NLP) that aims to condense large volumes of text into shorter, more manageable versions. This method involves selecting the most relevant sentences or phrases from the source text and combining them to create a summary that accurately conveys the essential information and main ideas of the source material (Narayan et al., 2018).

Abstractive Text Summarization is an advanced text summarization approach that employs NLP techniques to generate concise sentences that accurately convey the main ideas of the original text. This technique can benefit various domains where decision-makers require a rapid understanding of a document's primary points. Abstractive summarization can produce more coherent and efficient documents by eliminating redundancy and repetition. Unlike extractive text summarization, which selects and combines existing sentences or phrases, abstractive text summarization generates new and concise sentences, making it more versatile and flexible (Gupta and Gupta, 2019).

Hybrid Text Summarization combines extractive and abstractive text summarization strengths, resulting in a robust approach for condensing large volumes of text into shorter, more understandable versions. This technique minimizes word repetition and enhances the model's accuracy, necessitating ongoing refinement and experimentation to fine-tune the system and optimize its performance (Yadav et al., 2022).

2 Related Work

A recent study presented an NLP-based approach to generate business meeting summaries (Jha, Aryan et al., 2022). This research proposed a methodology employing various NLP techniques, such as Named Entity Recognition (NER), to identify crit-

Proceedings of Recent Advances in Natural Language Processing, pages 1274–1281
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_135

ical entities. Moreover, the authors utilized the "TextRank" algorithm, based on "PageRank", to rank meaningful sentences and generate summaries according to the sentence rankings. This proposed methodology belongs to the extractive text summarization category. The approach demonstrated promising results in extracting vital information from business meetings and generating summaries that capture the meetings' main ideas.

The application of NLP techniques for summarizing text data, including a transcribed speech from meetings or extracting critical details from articles, has increased interest. Another recent study (Agrawal et al., EasyChair, 2021) explores the topic of summarizing meeting transcripts from Google Meet. This study investigates the effectiveness of various NLP models for summarizing transcripts and compares several models using metrics such as ROUGE. The study offers insights into the performance of different NLP models for extractive and abstractive summarization tasks.

Building upon the insights from these studies, our proposed methodology introduces an enhanced TextRank approach using Cosine similarity for n-grams and fine-tuning hyperparameters. By addressing various pre-processing states, fine-tuning of TextRank, an intended combination of NLP summarization models into a hybrid model, and calculating the evaluation metrics using ROUGE scores widely used in previous research, to ensure a fair comparison with existing methods. We aim to improve extractive summarization's overall performance and accuracy by taking these steps. The reviewed literature provides a solid foundation for our proposed methodology, as it leverages state-of-the-art NLP techniques and insights gained from previous research, such as using TextRank to achieve the highest accuracy.

3 Methodology

Our methodology employs a TextRank algorithm enhanced with Cosine similarity for n-grams and fine-tuned hyperparameters to achieve optimal performance. This approach consists of four critical stages: preprocessing, fine-tuning TextRank, generating the summary, and evaluating the results using ROUGE scores, as shown in Figure 1.

3.1 TextRank Algorithm

TextRank is an unsupervised, graph-based algorithm for extractive summarization (Mihalcea and Tarau, 2004). Inspired by Googpreprocessing, it constructs a graph of sentences and calculates their importance based on connections to other sentences.

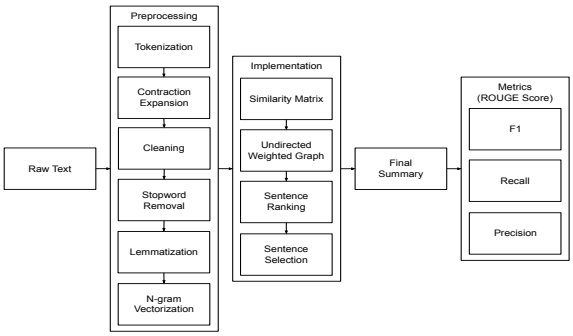

Figure 1: Methodology Flowchart

3.1.1 Cosine Similarity

Cosine similarity is a vector-based similarity measure that calculates the cosine of the angle between two vectors (Li and Han, 2013). In our implementation, we compute the cosine similarity between pairs of n-grams vectors. This similarity measure accounts for the frequency or importance of elements in the sets, making it more robust and flexible and allowing for a more accurate sentence comparison. The mathematical equation for the cosine similarity is represented as follows:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \quad (1)$$

3.1.2 Jaccard Similarity

The Jaccard similarity is a statistical used for comparing the similarity and diversity of sample sets. In the context of text summarization, we compute the Jaccard similarity between pairs of word sets derived from sentences. This set-based measure effectively captures semantic similarity by considering the shared vocabulary between sentences. It doesn't account for the frequency of words, emphasizing the unique shared and total elements. The mathematical equation for Jaccard similarity is represented as follows:

$$\text{Jaccard Similarity}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

Where:

- $|A \cap B|$ is the size of the intersection of sets A and B.
- $|A \cup B|$ is the size of the union of sets A and B.

3.1.3 Dice Similarity

Dice similarity is a statistical measure used for evaluating the similarity between two sets. It is particularly used in text analysis, where sets of words derived from sentences are compared. The Dice coefficient is calculated as twice the size of the intersection of sets, divided by the total size of both sets. This measure is similar to the Jaccard index but emphasizes sets' intersection. The mathematical equation for the Dice similarity is represented as follows:

$$\text{Dice Similarity}(A, B) = \frac{2|A \cap B|}{|A|+|B|} \quad (3)$$

Where:

- $|A \cap B|$ is the size of the intersection of sets A and B.
- $|A|$ and $|B|$ are the sizes of set A and set B, respectively.

3.2 Undirected Weighted Graph

In this section, we discuss the formulation of an undirected weighted graph, a pivotal step in the TextRank algorithm. Each sentence in the text under consideration is represented as a node in this graph. The edges that link these nodes carry a weight representing the similarity between sentences, as determined by a chosen similarity measurement function (Mihalcea, 2004).

The Cosine similarity is a measure based on the cosine of the angle between two vectors, in this context, the term-frequency vectors of two sentences. Jaccard similarity quantifies the proportion of shared terms to the total unique terms in both sentences. Dice similarity also considers shared terms but calculates the ratio to the average size of both sentences.

The graph construction involves each pair of sentences contributing an edge, the weight of which is determined by their similarity score according to the chosen metric. Consequently, more similar sentences will have a stronger connection in the graph, as reflected by higher edge weights.

The resulting undirected weighted graph forms the basis for applying the PageRank algorithm.

The concept is illustrated in Figure 2, where nodes (S_1, S_2, S_3, and S_4) correspond to sentences, and edges connecting them depict the relationship between these sentences. The weight labels $w_{i,j}$ represent the similarity scores between sentences i and j according to the chosen similarity metric.

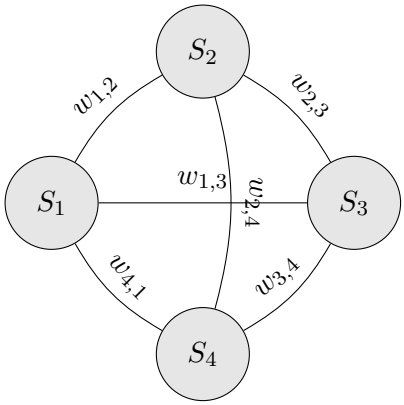

Figure 2: Undirected Weighted Graph

This graph-based text representation supports exploring inter-sentence relationships, which lies at the heart of the TextRank approach for extractive text summarization. Our experimental course with different similarity metrics aims to optimize this relationship exploration further and subsequently improve the summarization quality.

3.3 PageRank Algorithm

PageRank algorithm is a highly influential method developed by Page et al. (Page et al., 1999). The primary function of the PageRank algorithm is to compute the relative importance of nodes within a graph. It achieves this by incorporating an adjustable damping factor, which modulates the likelihood of arbitrary node transitions. This, in effect, mimics the actions of a web surfer arbitrarily transitioning between different web pages.

To achieve practical and efficient implementation of the PageRank algorithm, we utilized the NetworkX library. NetworkX is a comprehensive Python library that creates, manipulates, and investigates complex networks. Notably, it extends beyond the mere creation of networks to facilitate the computation of various network properties, such as the PageRank scores. In this study, NetworkX enabled us to transform our sentences into an interconnected network and apply the PageRank algorithm to the resultant web.

We calculated the PageRank scores of sentences using an iterative equation as provided by the NetworkX library:

$$PR^{(k+1)}(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR^{(k)}(p_j)}{L(p_j)} \quad (4)$$

In this equation, $PR^{(k+1)}(p_i)$ represents the PageRank of sentence p_i at iteration $k+1$, and d denotes the damping factor, an adjustable parameter that controls the probability of random jumps between nodes. N is the total number of sentences, and $M(p_i)$ signifies the set of sentences linking to p_i. Lastly, $L(p_j)$ represents the count of outbound links from sentence p_j. It is noteworthy that higher PageRank scores indicate more significant sentences, which are then included in the resultant summary. Using NetworkX in our approach allowed us to exploit the power of network analysis in the domain of extractive text summarization, making this study a multi-disciplinary endeavor.

3.3.1 Sentence Selection

Based on their PageRank scores, sentences are ranked (Goldstein et al., 1999), and then the top k penalties to include in the summary are selected. The number of sentences (k) is determined by a predefined percentage of the total sentences in the input text. Using a threshold-based sentence selection strategy, the method generates more accurate summaries that include only the most important sentences.

3.4 Summary Construction

Final summaries are formed by concatenating the selected sentences, ensuring the output is contextually relevant.

3.5 ROUGE Score

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is an important metric because it evaluates text summarization techniques' effectiveness (Lin, 2004). It measures the similarity between the machine-generated and reference summaries based on the number of overlapping n-grams. We tested our resumes with the most common use n-gram lengths 1 (unigrams), 2 (bigrams), and L (longest common subsequence).

3.5.1 Recall

The recall is the proportion of overlapping n-grams in the reference summary that is also present in the machine-generated summary. It is defined as:

$$Recall = \frac{Number\ of\ overlapping\ n-grams}{n-grams\ in\ reference\ summary} \quad (5)$$

3.5.2 Precision

Precision is the proportion of overlapping n-grams in the machine-generated summary also present in the reference summary. It is defined as:

$$Precision = \frac{overlapping\ n-grams}{n-grams\ in\ final\ summary} \quad (6)$$

3.5.3 F1-score

The F1-score is the harmonic mean of precision and recall. It balances the trade-off between precision and recall, providing a single metric for comparing summaries. The F1-score is defined as:

$$F1-score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7)$$

4 Experimental Framework

Our overall goal in this experiment is to gather critical key points from data. Therefore, we choose an extractive approach over abstractive and hybrid models. Extractive summarization methods identify and select the most important sentences from the source text, ensuring the critical information is preserved in the summary. This is particularly useful in professional settings where maintaining the accuracy and relevance of communication is crucial.

4.1 Dataset

In our research, we utilized the comprehensive BBC News Summary dataset. This dataset incorporates 2,225 documents, divided into five categories: Business, Entertainment, Politics, Sports, and Tech. The Business category contributes 510 articles, Entertainment presents 386 articles, Politics offers 417 articles, Sports provides 511 articles, and Tech supplies 401 articles. The diversity of these categories facilitates testing our model's performance across various subjects, certifying that our summarization method is adaptable and relevant in numerous contexts.

The dataset also provides fascinating insights into the average number of sentences across categories: Business features an average of 15.66 sentences, Entertainment averages 16.35 corrections, Politics comes in at 20.90 sentences, Sports averages 17.07 sentences, while Tech leads with an average of 24.05 penalties.

The balanced distribution of the dataset and its real-world applicability ensure the model's versatility in managing different content types. With an

extensive compilation of documents accompanied by their human-generated summaries, the dataset offers a fitting framework for comprehensive evaluation and benchmarking.

4.2 Similarity Matrices

In this study, we employed the top three similarity measures - Cosine, Jaccard, and Dice - to evaluate the performance of the TextRank algorithm in the context of extractive summarization. We aimed to investigate which similarity measure leads to the most accurate summaries according to the ROUGE metrics (Recall, Precision, and F1 score). After implementing TextRank using each similarity measure, we observed that Cosine similarity outperformed both Jaccard and Dice regarding ROUGE scores.

Since cosine normalizes the vectors by their magnitude, it is less sensitive to the difference in lengths of the vectors (i.e., the number of words or tokens in the sentences). This property allows the Cosine similarity measure to assess the similarity between sentences better, even when they differ in length or word count.

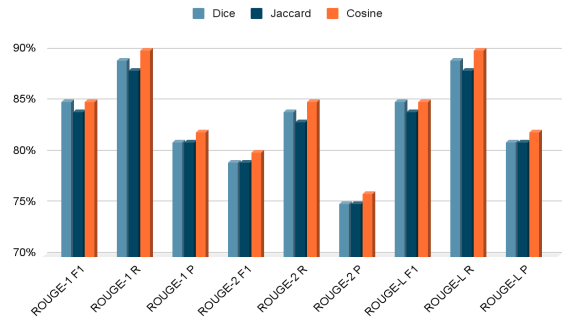

Figure 3: Mean ROUGE Scores by Similarity

On the other hand, Jaccard and Dice similarity measures are based on the ratio of the size of the intersection of the sets to their union or the average of their sizes, respectively. These measures can be more sensitive to differences in sentence lengths and word counts, which might lead to less accurate comparisons between sentences. Consequently, they may not be as effective as Cosine similarity in capturing the semantic similarity between sentences.

The superior performance of Cosine similarity can be attributed to its ability to capture the underlying semantic relationship between sentences more effectively than Jaccard and Dice similarity measures, as shown in Figure 3. This is particularly important in extractive summarization, where the goal is identifying and selecting the most relevant and informative sentences from the original text. By leveraging the strengths of Cosine similarity, the TextRank algorithm can better identify and rank sentences that capture the essence of the source document, leading to more accurate and coherent summaries.

4.3 Tuning Hyperparameters

Our method allows us to customize the percentage of sentences to include in the summary, the rates of sentences, the n-gram range vectorization, and the dampening factor. This flexibility enables the algorithm to adapt to different documents and use cases, ensuring the generated summaries are relevant and valuable.

4.3.1 Percentages of Sentences

We experimented with different values for the summary percentage. As you can see in Figure 4 when using higher rates than 50%, we observed that the precision scores decreased while recall increased. This is because as more sentences are included in the summary, it becomes more likely that non-relevant information will be introduced, leading to a drop in precision. Conversely, recall improves as more content from the original text is covered. On the other hand, when lowering the percentage, the opposite occurs.

Our optimal scores were between 45% and 50%. When calculating the average reference summaries in the entire BBC News Summary data set, we found it to be 45%. However, we stuck with 50% since it was the optimal F_1 score and maintained an over better recall score, which is important in maintaining the key details in data collection.

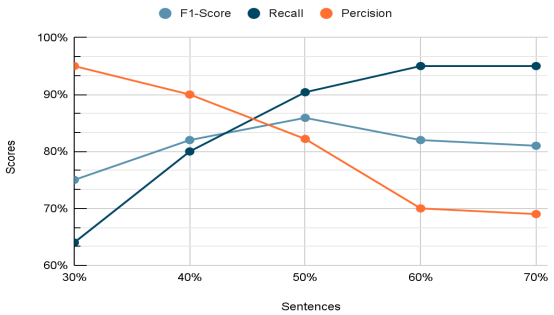

Figure 4: ROUGE-1 Scores by Percentage of Sentences

4.3.2 N-Gram

We tested various n-grams to determine the best configuration for our TextRank-based summarization model. We conducted experiments with n-grams ranging from unigrams (1-1) to 8-grams (1-8). We stopped at 8-grams because the results stayed the same. Our primary goal was to find the optimal n-gram configuration that would yield the highest summarization performance.

Our results indicate that unigrams outperformed all tested n-grams. Increasing the n-gram range resulted in a consistent decrease in performance, suggesting that higher n-grams could not capture the necessary information for accurate summarization. Therefore, our findings indicate that unigrams are the optimal n-gram configuration for our TextRank-based summarization model, allowing it to capture the most relevant information and produce more accurate summaries.

Our findings concluded that unigrams were the optimal n-gram configuration for our TextRank-based summarization model. Using unigrams allowed the model to capture the most relevant information from the text, leading to more accurate summaries.

4.3.3 Dampening Factor

For each dampening factor, the F_1 scores of the generated summaries were measured using the Rouge-1 metric. The F_1 scores increased consistently as the dampening factor increased, indicating that the outlines became more accurate and aligned with the reference summaries. The improvement in F_1 scores continued until the dampening factor reached 0.95, where the optimal performance was achieved.

After the dampening factor reached 0.95, there were no further improvements in the F_1 scores, suggesting that the optimal setting for the dampening factor in this experiment is 0.95. Using this optimal setting, the algorithm could effectively generate high-quality extractive summaries, balancing precision and recall.

Fine-tuning the TextRank algorithm with dampening factors significantly enhanced the quality of the generated summaries. By carefully selecting the optimal dampening element, n-gram range, and similarity measure, the algorithm became more efficient in capturing the most relevant and essential information from the source text. This fine-tuning allowed for a better balance between precision and recall, resulting in summaries that closely matched the reference summaries. These adjustments led to a more accurate and coherent extractive summarization that effectively condensed the main ideas from the original content.

5 Results Analysis and Discussion

TextRank - Extractive			
	ROUGE-1	ROUGE-2	ROUGE-L
Precision	0.70	-	-
Recall	0.8581	-	-
F_1 Score	0.7594	-	-

NLTK - Extractive			
	ROUGE-1	ROUGE-2	ROUGE-L
Precision	0.731	0.759	0.710
Recall	0.767	0.701	0.769
F_1 Score	0.713	0.651	0.732

Enhanced TextRank - Extractive			
	ROUGE-1	ROUGE-2	ROUGE-L
Precision	0.822	0.767	0.820
Recall	0.904	0.859	0.903
F_1 Score	0.859	0.808	0.858

Table 1: Comparison of extractive models

5.1 Evaluating Extractive Models: Our Approach vs. Conventional TextRank

The principal extractive summarization model adopted in our study is TextRank, inspired by the PageRank algorithm (Jha, Aryan et al., 2022). Our approach enhances TextRank's effectiveness by incorporating advanced preprocessing methods, a refined similarity measure, and optimizing the damping factor.

1. **Advanced Preprocessing:** Our approach uses a combination of sophisticated natural language processing libraries, including NLTK and Spacy, for sentence tokenization and lemmatization, which are critical for maintaining sentence-level semantics. Using a predefined contractions dictionary and regular expressions facilitates consistent text formatting through contractions expansion. In addition, noise reduction in the textual data is achieved by removing stopwords and filtering sentences based on length.

2. **Refined Similarity Measure:** Using Scikit-learn's feature extraction tools for n-gram vec-

torization, and the computation of cosine similarity, we create an adjacency matrix that more accurately reflects sentence connections. This enhanced similarity measure, which accounts for the frequency and importance of elements in the sets, improves sentence comparison and the subsequent construction of the sentence graph.

3. **Damping Factor Optimization:** The application of the NetworkX library allows for fine-tuning the damping factor in the PageRank algorithm, a key parameter that controls the probability of random jumps between nodes. These optimization steps better balance precision and recalls in the summarization process.

Our approach achieves superior performance metrics through these refinements over the conventional TextRank model. With a ROUGE-1 F_1 score of 0.859, our model outperforms the traditional TextRank score of 0.7594. Moreover, it records a ROUGE-2 F_1 score of 0.808 and a ROUGE-L score of 0.858, testifying its ability to generate more coherent, structured, and contextually preserved summaries. The higher F_1 scores across all ROUGE metrics reflect the model's strength in producing accurate and informative summaries, marking its broad applicability in various scenarios.

5.1.1 Comparison with EasyChair NLTK Model

The NLTK model is an extractive text summarization method that leverages the Natural Language Toolkit (NLTK), a powerful Python library for computational linguistics. This approach to summarization focuses on selecting top-ranked sentences from the original text to generate the summary. The methodology involves several steps: data preprocessing, tokenization, generating a word frequency table, and sentence scoring based on word frequencies (Agrawal et al., EasyChair, 2021).

The preprocessing phase aims to clean the input text from redundant information and remove stop words. Following this, the reader is tokenized into words and sentences. The word frequency table is then generated to identify the most critical comments in the text, which will be used to calculate sentence scores. The NLTK model selects sentences with the highest scores to form the final summary. While this approach is straightforward, it often falls short in capturing complex relationships between words and maintaining the overall coherence and context of the original text.

The improvements and optimizations in our research approach to TextRank allow it to achieve a ROUGE-1 F_1 score of 0.859 compared to the existing NLTK model's 0.651. The higher F_1 score highlights our model's ability to balance precision and recall, generating informative and accurate summaries essential for various applications.

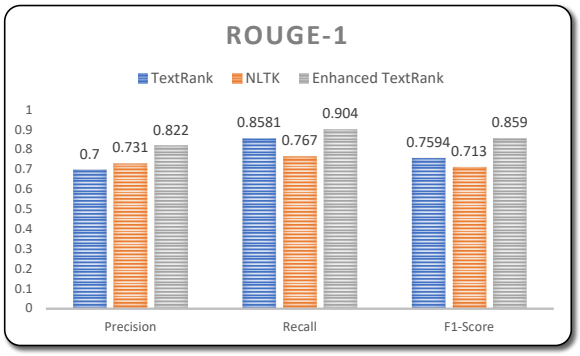

Figure 5: Comparison of ROUGE-1 scores

Our research approach to TextRank outperforms the NLTK extractive method. The superiority of our research approach to TextRank can be attributed to the advanced preprocessing techniques, tokenization, word embeddings, and similarity measures we employ. By incorporating these features, our system can produce high-quality summaries that effectively represent the main ideas and structure of the original text, making it a more suitable choice for various applications that demand accurate and informative summaries.

In conclusion, our research approach to TextRank significantly improves the existing NLTK model and offers competitive performance. The enhancements in preprocessing, tokenization, word embeddings, and similarity measures enable our model to generate high-quality summaries that accurately represent the main ideas and structure of the original text. As a result, our approach is a more viable option for various applications requiring coherent and contextually accurate summaries.

6 Conclusions

This study proposes a refined approach to the TextRank model for extractive text summarization. Our methodology outperforms the existing TextRank method (Jha, Aryan et al., 2022) and the NLTK extractive model (Agrawal et al., EasyChair, 2021) on various ROUGE metrics.

Acknowledgments

The authors thank the CSULB College of Engineering and the CSULB Department of Computer Engineering and Computer Science for their support.

References

Yash Agrawal, Atul Thakre, Tejas Tapas, Ayush Kedia, Yash Telkhade, and Vasundhara Rathod. EasyChair, 2021. Comparative analysis of nlp models for google meet transcript summarization.

Md Siam Ansary. 2021. A hybrid approach for automatic extractive summarization. In *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pages 11–15.

Yazid Bounab, Joshua Muyiwa Adeegbe, and Mourad Oussalah. 2019. Towards storytelling automatic textual summerized. In *Conference of Open Innovations Association, FRUCT*, volume 25, pages 434–438. FRUCT Oy.

Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 121–128.

Som Gupta and Sanjai Kumar Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Jha, Aryan, Temkar, Sameer, Hegde, Preetam, and Singhaniya, Navin. 2022. Business meeting summary generation using nlp. *ITM Web Conf.*, 44:03063.

Baoli Li and Liping Han. 2013. Distance weighted cosine similarity measure for text classification. In *Intelligent Data Engineering and Automated Learning–IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, October 20-23, 2013. Proceedings 14*, pages 611–618. Springer.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, pages 74–81.

Rada Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL interactive poster and demonstration sessions*, pages 170–173.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. In *Technical report, Stanford InfoLab*.

Arun Kumar Yadav, Amit Singh, Mayank Dhiman, Vineet, Rishabh Kaundal, Ankit Verma, and Divakar Yadav. 2022. Extractive text summarization using deep learning approach. *International Journal of Information Technology*, 14.