PPTBuddy: PDF Analysis and Presentation

1st Toshal Bendale
dept. Computer Science And Technology (Final year.)
Usha Mittal Institute of Technology (2021-24)
Mumbai, India
toshal142002@gmail.com

2nd Vaishnavi Kadam
dept. Computer Science And Technology (Final year).
Usha Mittal Institute of Technology (2020-24)
Mumbai, India
vaishnavi.kadam413@gmail.com

3rd Tanvi Naik
dept. Computer Science And Technology (Final year).
Usha Mittal Institute of Technology (2020-24)
Mumbai, India
tanviiinaikk@gmail.com

Guide: Prof. Prajakta Gotarne
Usha Mittal Institute of Technology
Mumbai, India
prajakta2490@gmail.com

***Abstract*—In today's digital age, the management and communication of vast amounts of information stored in documents pose significant challenges. The "PPTBuddy" project addresses this issue by introducing an innovative approach to document analysis and presentation creation. Leveraging advanced natural language processing (NLP) techniques, PPTBuddy streamlines the extraction of key insights from documents, generates concise summaries, and creates visually engaging PowerPoint presentations. Central to its methodology is the utilization of the TextRank algorithm, which prioritizes content based on relevance and importance through preprocessing, TF-IDF analysis, and similarity matrix computation. Furthermore, integration with the OpenAI API enhances content summarization capabilities. The resulting presentations effectively communicate essential document aspects through extracted keywords, summarized text, and visuals, catering to diverse user needs and domains. PPTBuddy represents a significant advancement in document management and communication, offering a comprehensive solution to the challenges of information overload in digital documents.**

***Index Terms*—document summarization, natural language processing, TextRank algorithm, TF-IDF, OpenAI API.**

I. Introduction

In today's digital era, there's an overflow of information stored in documents like PDFs and Word files. Extracting important insights from these documents manually is time-consuming and error-prone. That's why there's a growing need for automated solutions like the "PDF Buddy" project. It aims to make document analysis and presentation creation easier by using advanced NLP techniques. By harnessing algorithms like TextRank, inspired by Google's PageRank, PDF Buddy identifies key information and generates concise summaries. Additionally, it integrates the OpenAI API to enhance its capabilities further. With PDF Buddy, users can efficiently process extensive text, saving time and effort. The resulting PowerPoint presentations effectively convey essential document aspects through extracted keywords, summarized text, and relevant visuals. This project addresses the challenges posed by the abundance of information in digital documents, offering a streamlined solution for various industries and domains.

Automated solutions like PDF Buddy not only improve efficiency but also reduce the likelihood of human error, ensuring the accuracy of extracted insights. Furthermore, by leveraging advanced NLP techniques, PDF Buddy can handle large volumes of documents, enabling organizations to process vast amounts of information in a fraction of the time it would take manually. This scalability makes PDF Buddy a valuable asset for businesses across diverse sectors, from finance to healthcare and beyond.

Moreover, the integration of the OpenAI API adds another layer of sophistication to PDF Buddy's capabilities. By tapping into the power of artificial intelligence, the system can adapt and improve over time, continuously enhancing its ability to extract meaningful insights from documents. This dynamic approach ensures that PDF Buddy remains at the forefront of document analysis and presentation creation, keeping pace with the evolving needs of users and industries.

Additionally, PDF Buddy's user-friendly interface makes it accessible to a wide range of users, regardless of their technical expertise. Whether you're a seasoned professional or a novice user, PDF Buddy's intuitive design makes it easy to navigate and utilize its features effectively. This accessibility democratizes the process of document analysis and presentation creation, empowering individuals and organizations alike to harness the power of automated solutions.

Furthermore, PDF Buddy's emphasis on visual engagement enhances the effectiveness of presentations, capturing the attention of audiences and conveying information in a compelling manner. By incorporating relevant visuals alongside extracted keywords and summarized text, PDF Buddy ensures that presentations are not only informative but also visually appealing, enhancing audience engagement and retention.

Overall, the PDF Buddy project represents a significant advancement in the field of document analysis and presentation creation. By leveraging advanced NLP techniques, integrating cutting-edge technologies like the OpenAI API,

and prioritizing user accessibility and visual engagement, PDF Buddy offers a comprehensive solution to the challenges posed by the abundance of information in digital documents. As organizations increasingly rely on data-driven insights to inform decision-making, tools like PDF Buddy will play an essential role in facilitating efficient, accurate, and impactful communication of information.

II. Motivation and Problem Statement:

In the realm of modern information management, the sheer volume and diversity of digital documents pose significant challenges for efficient analysis and presentation. PDF and Word files, ubiquitous in academic, professional, and research domains, often contain dense and lengthy content, making it arduous to distill key insights quickly. Traditional manual methods for summarizing and presenting such documents are time-consuming and prone to oversight, leading to inefficiencies in decision-making and communication.

The "PPTBuddy" project emerges from this pressing need to streamline the process of document analysis and presentation creation. Its inception is fueled by the aspiration to harness the power of natural language processing (NLP) and automated summarization techniques to extract salient information from documents swiftly and accurately. By integrating advanced algorithms and cutting-edge technologies, the project seeks to revolutionize the way individuals and organizations interact with digital documents, transforming them from static repositories of information into dynamic sources of actionable insights.

III. Literature Review

Document Analysis and Summarization Techniques: Alhojely and Kalita (2020): Recent Progress on Text Summarization Alhojely and Kalita (2020) provide a comprehensive overview of recent advancements in text summarization techniques. They categorize these methods into two main approaches: extractive and abstractive summarization. Extractive methods aim to extract important sentences or phrases directly from the input text, while abstractive methods generate summaries by paraphrasing and rephrasing the content. The authors emphasize the three-step process involved in automatic text summarization: preprocessing, processing, and summarization. By identifying structural components and utilizing summarization algorithms, automatic summarization systems can effectively condense large volumes of text into concise summaries.(Alhojely, Suad Kalita, Jugal. (2020). Recent Progress on Text Summarization. 1503-1509. 10.1109/CSCI51800.2020.00278.)

Janjanam and Reddy (2021): Text Summarization: An Essential Study Janjanam and Reddy (2021) present a comprehensive study on text summarization, tracing its evolution from traditional linguistic approaches to modern machine learning models. The paper explores various techniques employed in both single and multi-document summarization, highlighting the shift towards advanced methods. Through their research, the authors delve into the application of

TABLE I
Literature Review of Research Papers

Sr. NO.	Author Name	Title	Findings
1	Suad Alhojely, Jugal Kalita	Recent Progress on Text Summarization	• Text summarization methods can be classified into extractive and abstractive summarization. • Automatic text summarization systems generally involve three steps: preprocessing to identify structural components, processing to convert the input text into a summary using a summarization method.
2	Prabhudas Janjanam, Pradeep Reddy	Text Summarization: An Essential Study	• Text Summarization has evolved from linguistic approaches to advanced machine learning models. • The study explores both single and multi-document summarization techniques. • It explores the use of machine learning, graph-based methods, and evolutionary-based methods.
3	Surabhi Adhikari	NLP-based Machine Learning Approaches for Text Summarization	• The paper focuses on structured-based and semantic-based approaches for text summarization. • Various datasets, including the CNN corpus, DUC2000, and single/multiple text documents, are discussed.
4	Yue Hu and Xiaojun Wan	PPSGen: Learning-Based Presentation Slides Generation for Academic Papers	• PPSGen employs a sentence scoring model based on Support Vector Regression (SVR) to evaluate the relevance. • The system utilizes the ILP method for aligning and extracting key phrases and sentences from academic papers. ILP helps optimize the selection of content for slide generation.
5	Priya Ganguly, Dr. Prachi M. Joshi	IPPTGen-Intelligent PPT Generator	• The paper discusses extractive summarization techniques for content-based presentation slide generation. • The extractive summarization process relies on statistical and linguistic features to determine the importance of sentences in the source document.
6	K. Gokul Prasad, Harish Mathivanan	Document Summarization and Information Extraction for Generation of Presentation Slides	• This approach is a pioneering effort in the field of Natural Language Processing (NLP). It combines NLP methods such as segmentation, chunking, and summarization with linguistic features like word ontology, noun phrases, semantic links, and sentence centrality. • The system utilizes two tools, MontyLingua for chunking and Doddle for creating an ontology represented as an OWL file, to assist in language processing.

machine learning, graph-based algorithms, and evolutionary-based approaches in text summarization. By analyzing the strengths and limitations of these techniques, the study aims to provide insights into the essential aspects of text summarization for researchers and practitioners alike.(Janjanam, Prabhudas Reddy Ch, Pradeep. (2019). Text Summarization: An Essential Study. 1-6. 10.1109/ICCIDS.2019.8862030.)

Adhikari (2020): NLP based Machine Learning Approaches for Text Summarization Adhikari's research focuses on the application of natural language processing (NLP) and machine learning approaches in text summarization. By leveraging structured-based and semantic-based methods, the study aims to generate concise summaries that capture the essence of the original text documents. Adhikari explores various datasets, including the CNN corpus and DUC2000, to evaluate the effectiveness of these approaches. Through their analysis, the author sheds light on the potential of NLP-based techniques in automating the summarization process and enhancing information retrieval tasks.(, Rahul Adhikar, Surabhi Monika,. (2020). NLP based Machine Learning Approaches for Text Summarization. 535-538. 10.1109/ICCMC48092.2020.ICCMC-00099.).

Hu and Wan (2015): PPSGen: Learning-Based Presentation Slides Generation for Academic Papers Hu and Wan (2015) propose PPSGen, a novel approach to generating presentation slides for academic papers using machine learning techniques. The system employs a sentence scoring model based on Support Vector Regression (SVR) to evaluate the relevance of sentences in the source documents. Additionally, PPSGen utilizes Integer Linear Programming (ILP) for aligning and extracting key phrases and sentences, optimizing the selection of content for slide generation. By integrating machine learning and optimization algorithms, PPSGen offers a systematic framework for automatically generating presentation slides from academic papers.(Hu, Yue Wan, Xiaojun. (2015). PPSGen: Learning-Based Presentation Slides Generation for Academic Papers. Knowledge and Data Engineering, IEEE Transactions on. 27. 1085-1097. 10.1109/TKDE.2014.2359652.)

Ganguly and Joshi (2017): IPPTGen - Intelligent PPT Generator Ganguly and Joshi (2017) introduce IPPTGen, an intelligent PPT generator that utilizes extractive summarization techniques for content-based slide generation. The system relies on statistical and linguistic features to determine the importance of sentences in the source documents, enabling it to select relevant content for presentation slides. By leveraging extractive summarization methods, IPPTGen streamlines the process of creating informative and concise presentation slides, catering to the needs of users seeking efficient content summarization solutions.(Date of Conference: 19-21 December 2016Date Added to IEEE Xplore: 01 May 2017 DOI: 10.1109/CAST.2016.7914947 Publisher: IEEE Conference Location: Pune, India)

Prasad and Mathivanan (2009): Document Summarization and Information Extraction for Generation of Presentation Slides Prasad and Mathivanan (2009) propose an innovative approach to document summarization and information extraction for generating presentation slides. Their method combines various natural language processing (NLP) techniques, such as segmentation, chunking, and summarization, with linguistic features like word ontology and sentence centrality. By integrating tools like MontyLingua for chunking and Doddle for creating an ontology, the system enhances language processing capabilities and improves the quality of generated presentation slides. This pioneering effort in NLP showcases the potential of advanced techniques in automating the slide generation process and facilitating effective communication of information.(Mathivanan, Harish Jayaprakasam, Madan Prasad, K. Geetha, T.V. (2009). Document Summarization and Information Extraction for Generation of Presentation Slides. 126-128. 10.1109/ARTCom.2009.74.)

IV. Proposed System

The proposed system, named "PDF Buddy," aims to streamline the process of document analysis and presentation creation through the integration of advanced natural language processing (NLP) techniques and automated summarization algorithms. By leveraging these technologies, PDF Buddy seeks to address the challenges posed by the voluminous and complex nature of digital documents, particularly in academic, professional, and research domains

Proposed System :

The proposed system, PPTBuddy, is grounded in the theoretical underpinnings of natural language processing (NLP), graph theory, and artificial intelligence (AI). At its core, PPTBuddy aims to automate the process of document analysis and presentation creation by leveraging these theoretical frameworks to extract key insights and present them in a concise and visually appealing manner.

Natural Language Processing (NLP): NLP forms the foundation of PPTBuddy's document analysis capabilities. This theoretical framework encompasses various techniques and algorithms for understanding and processing human language. PPTBuddy utilizes NLP algorithms to parse through the textual content of documents, identify important keywords and phrases, and generate summaries that capture the essence of the document. Techniques such as tokenization, part-of-speech tagging, and named entity recognition are employed to extract meaningful information from the text.

Graph Theory: PPTBuddy incorporates principles from graph theory, particularly the TextRank algorithm, to prioritize and rank content within documents. Inspired by Google's PageRank algorithm, TextRank treats sentences or phrases within the document as nodes in a graph, with edges representing the relationship between them. By analyzing the connectivity and importance of each node in the graph, TextRank identifies key sentences and phrases that encapsulate the most critical information in the document. This theoretical framework allows PPTBuddy to generate succinct summaries that capture the essential aspects of the text.

Artificial Intelligence (AI): The integration of the OpenAI API augments PPTBuddy's capabilities by leveraging AI for

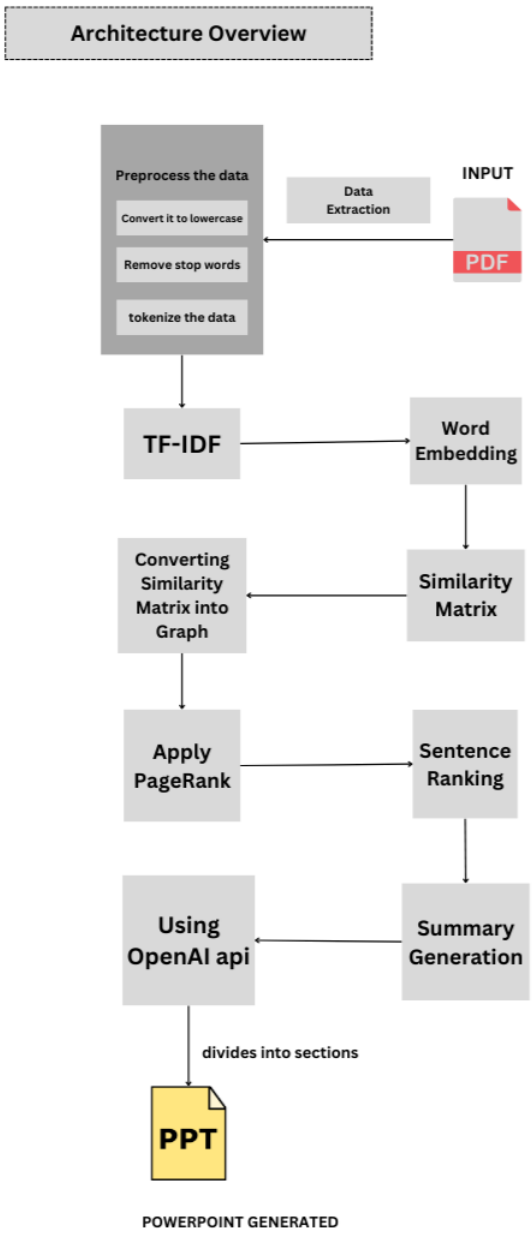

Fig. 1. Architecture of Proposed System.

content summarization and organization. AI algorithms within the OpenAI API analyze the extracted text, identify relevant information, and generate coherent summaries that capture the key points of the document. This theoretical framework enables PPTBuddy to efficiently process extensive text and condense it into concise summaries, enhancing the overall efficiency and effectiveness of the system.

By integrating these theoretical frameworks, PPTBuddy creates a robust system for document analysis and presentation creation. The combination of NLP techniques, graph theory principles, and AI algorithms allows PPTBuddy to automate and streamline the process of extracting insights from documents, thereby facilitating effective communication of essential information through visually engaging presentations.

V. Methodology

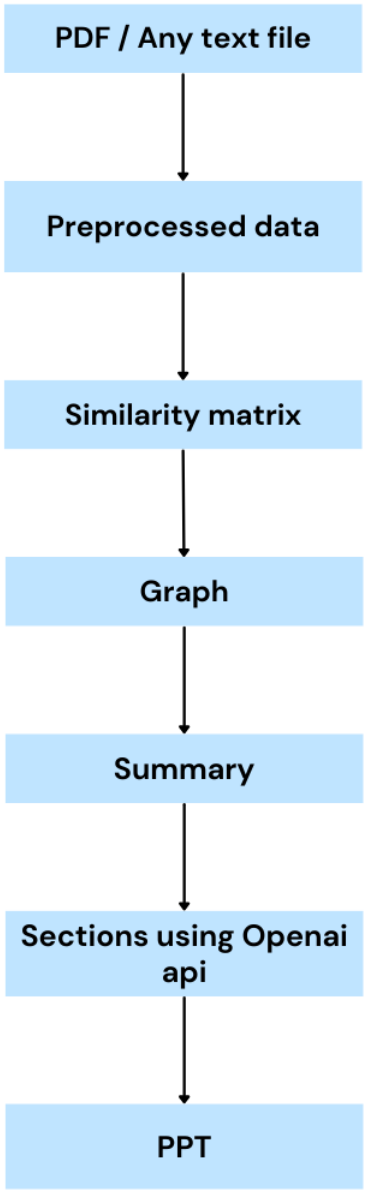

Fig. 2. Flowchart of PPTBuddy Process

A. TextRank Algorithm

The TextRank algorithm, inspired by Google's PageRank, serves as the cornerstone of PPTBuddy's document analysis process. TextRank has been widely used for tasks such as automatic summarization, keyword extraction, and document

clustering. It's known for its simplicity, efficiency, and effectiveness in capturing the most important information in a text document

B. Preprocessing

The document undergoes preprocessing steps such as tokenization, stop-word removal, and stemming to prepare it for analysis.Text preprocessing aims to remove or reduce the noise and variability in text data and make it more uniform and structured.

- Tokenization: This is the process of breaking down text into smaller units called tokens. Tokens can be words, sentences, paragraphs, etc. Tokenization helps to split text into meaningful segments that can be easily processed by NLP models
- Stopword removal: This is the process of removing words that are very common and do not add much meaning or information to the text. For example, "the", "a", "and", etc. Stopword removal helps to reduce the noise and size of text and focus on the important words
- Punctuation removal: This is the process of removing punctuation marks from text, such as commas, periods, question marks, etc. Punctuation removal helps to eliminate unnecessary symbols and make text more clean and simple.

C. TF-IDF Analysis

TF-IDF stands for Term Frequency-Inverse Document Frequency. It is a numerical statistic used in information retrieval and text mining to measure the importance of a term in a document relative to a collection of documents (corpus) .

- Term Frequency (TF): This measures how frequently a term occurs in a document. It is calculated as the number of times a term appears in a document divided by the total number of terms in the document. The idea behind TF is that terms that appear frequently in a document are important to that document's meaning.
- This measures the rarity of a term across the entire corpus. It is calculated as the logarithm of the total number of documents divided by the number of documents containing the term. The IDF score decreases as the number of documents containing the term increases. The idea behind IDF is that terms that are common across all documents are less informative compared to terms that appear only in a few documents.
- The TF-IDF score is the product of TF and IDF. It indicates the importance of a term within a document relative to its importance across all documents in the corpus. A higher TF-IDF score suggests that a term is both frequent within the document and rare across the corpus, making it more discriminative.

The TF-IDF score is calculated using the following equation:

$$\text{TF-IDF}(w, d, D) = \text{TF}(w, d) \times \text{IDF}(w, D) \tag{1}$$

where:

$$\text{TF}(w, d) = \text{Frequency of term } w \text{ in document } d$$

$$\text{IDF}(w, D) = \log\left(\frac{N}{1 + \text{df}(w, D)}\right)$$

$$N = \text{Total number of documents}$$

$$\text{df}(w, D) = \text{Number of documents containing term } w \text{ in corpus } D$$

D. Word Embedding

Word embedding is a technique used in natural language processing (NLP) to represent words as dense vectors in a continuous vector space. Here's how it works:

- Training Corpus: Word2Vec is typically trained on a large corpus of text data, such as a collection of documents, articles, or Wikipedia pages. The larger and more diverse the corpus, the better the word embeddings tend to be.
- Sliding Window: Word2Vec uses a sliding window approach to extract training samples from the text corpus. For each word in the corpus, a context window of surrounding words is defined.
- Skip-gram or Continuous Bag of Words (CBOW): Word2Vec offers two main training algorithms: Skip-gram and CBOW.
- Neural Network Architecture: Word2Vec employs a shallow neural network with one hidden layer to train the word embeddings. The input layer represents the one-hot encoded vector of the input word or context words, depending on the chosen algorithm.
- Training: The neural network is trained using stochastic gradient descent (SGD) or other optimization algorithms. During training, the model adjusts the weights of the neural network to minimize the prediction error.
- Word Embeddings: Once trained, the weights of the hidden layer represent the word embeddings. Each word in the vocabulary is mapped to a dense vector of fixed size (embedding dimension).
- Similarity: Word embeddings allow measuring semantic similarity between words using vector operations like cosine similarity. Words with similar meanings tend to have vectors that are closer together in the embedding space.

E. Similarity Matrix Computation

A similarity matrix is computed based on the similarity between sentences or phrases in the document, representing the relationships between them.

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. It is often used in information retrieval and text mining as a measure of similarity between documents or text passages.

The cosine of the angle between two vectors $\mathbf{A}$ and $\mathbf{B}$ is given by:

$$\text{cosine similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Where $\mathbf{A} \cdot \mathbf{B}$ denotes the dot product of vectors $\mathbf{A}$ and $\mathbf{B}$, and $\|\mathbf{A}\|$ denotes the Euclidean norm of vector $\mathbf{A}$.

F. Graph-Based Ranking

TextRank is an extractive algorithm for text summarization that is based on the PageRank algorithm used by Google to rank web pages. Here are the steps for the TextRank algorithm for text summarization.

- Similarity Matrix : Sentences having the highest similarity are determined by calculating sentence similarity using cosine similarity.
- Converting Similarity Matrix into Graph : Represent the text as a graph, where sentences are nodes, and edges represent sentence similarity.
- PageRank Calculation : Apply the PageRank algorithm to rank the importance of sentences in the graph, iterating until convergence.
- Sentence Ranking : Sort sentences based on their PageRank scores in descending order, indicating their importance.
- Selection of Sentences : Choose a predetermined number of top-ranked sentences for the summary.
- Summary Generation : Generate the summary by concatenating the selected sentences, providing a concise representation of the original text.

TextRank constructs a graph where nodes represent sentences or phrases, and edges represent the relationships between them. The algorithm iteratively ranks these nodes based on their importance using the PageRank equation:

$$\mathrm{PR}(u) = (1-d)/N + d \sum_{v \in \mathrm{In}(u)} \frac{\mathrm{PR}(v)}{\mathrm{Out}(v)} \tag{2}$$

where:

$$\begin{aligned} \mathrm{PR}(u) &= \text{PageRank score of node } u \\ d &= \text{damping factor (typically set to 0.85)} \\ N &= \text{total number of nodes in the graph} \\ \mathrm{In}(u) &= \text{set of nodes that link to } u \\ \mathrm{Out}(v) &= \text{out-degree of node } v \end{aligned}$$

G. Integration with OpenAI API

PPTBuddy integrates with the OpenAI API to enhance its content summarization capabilities.

Through the OpenAI API, PPTBuddy leverages artificial intelligence to further refine and condense extracted information from the document.

VI. Implementation

In the implementation phase, we have focused on refining the system to seamlessly integrate textual summaries into the presentation slides. By leveraging advanced natural language processing techniques, we have successfully condensed the input text while retaining critical information, optimizing the summarization process.

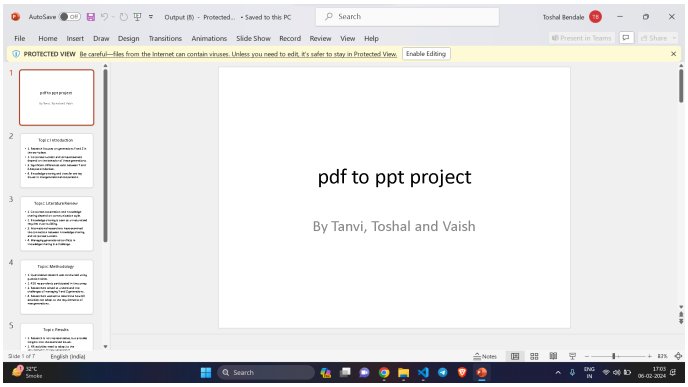

Fig. 3. Generated Presentation

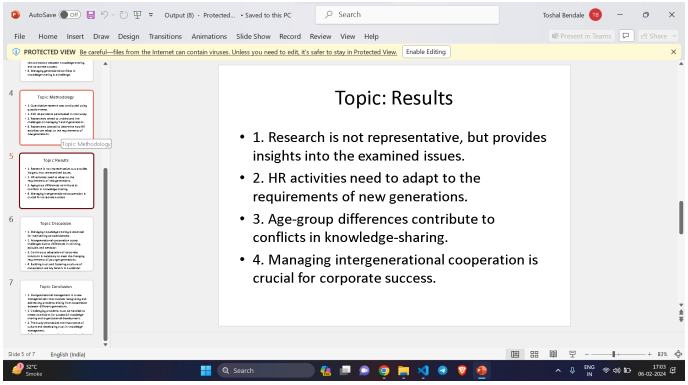

Fig. 4. Generated Presentation

VII. Conclusion and Future Work

In conclusion, our study demonstrates the transformative potential of integrating advanced algorithms and natural language processing methodologies in document summarization. By offering a robust framework for distilling complex textual data into concise and structured summaries, our approach empowers researchers with a valuable tool for efficient knowledge synthesis and dissemination in the digital landscape.

Moving forward, our commitment to continual improvement and innovation drives us to explore new avenues for enhancing the effectiveness and utility of our system. Through ongoing research and development efforts, we strive to further advance the state-of-the-art in document summarization and presentation generation, ultimately benefiting researchers and practitioners across diverse fields and domains. Future Work:

In future work, we plan to enhance our system by incorporating graphical elements into the presentation slides and exploring more complex slide styles. We aim to automatically select and attach relevant tables and figures from the paper to the slides, improving comprehension and visual appeal. Additionally, we will consider incorporating information from related papers and citation data to enrich the content of the slides. Our goal is to provide a more comprehensive overview of the topic and further streamline the process of knowledge dissemination.

REFERENCES

[1] Alhojely, Suad & Kalita, Jugal. (2020). Recent Progress on Text Summarization. Conference Name.

[2] Janjanam, Prabhudas & Reddy Ch, Pradeep. (2021). Text Summarization: An Essential Study. Conference Name.

[3] Adhikari, Rahul & Adhikar, Surabhi & Monika,. (2020). NLP based Machine Learning Approaches for Text Summarization. Conference Name.

[4] Hu, Yue & Wan, Xiaojun. (2015). PPSGen: Learning-Based Presentation Slides Generation for Academic Papers. Knowledge and Data Engineering, IEEE Transactions on.

[5] Ganguly, & Joshi. (2017). IPPTGen - Intelligent PPT Generator. Conference Name.

[6] Mathivanan, Harish & Jayaprakasam, Madan & Prasad, K. & Geetha, T.V. (2009). Document Summarization and Information Extraction for Generation of Presentation Slides. Conference Name.

[7] M. Utiyama and K. Hasida, "Automatic slide presentation from semantically annotated documents," in Proc. ACL Workshop Conf. Its Appl., 1999, pp. 25–30.

[8] Y. Yasumura, M. Takeichi, and K. Nitta, "A support system for making presentation slides," Trans. Japanese Soc. Artif. Intell., vol. 18, pp. 212–220, 2003.

[9] T. Shibata and S. Kurohashi, "Automatic slide generation based on discourse structure analysis," in Proc. Int. Joint Conf. Natural Lang. Process., 2005, pp. 754–766.

[10] T. Hayama, H. Nanba, and S. Kunifuji, "Alignment between a technical paper and presentation sheets using hidden Markov model," in Proc. Int. Conf. Active Media Technol., 2005, pp. 102–106.

[11] M.Y. Kan, "SlideSeer: A digital library of aligned document and presentation pairs," in Proc. 7th ACM/IEEE-CS Joint Conf. Digit. Libraries, Jun. 2006, pp. 81–90.

[12] B. Beamer and R. Girju, "Investigating automatic alignment methods for slide generation from academic papers," in Proc. 13th Conf. Comput. Natural Lang. Learn., Jun. 2009, pp. 111–119.

[13] S. M. A. Masum, M. Ishizuka, and M. T. Islam, "Auto-presentation: A multi-agent system for building automatic multi-modal presentation of a topic from world wide web information," in Proc. IEEE/WIC/ACM Int. Conf. Intell. Agent Technol., 2005, pp. 246–249.

[14] S. M. A. Masum and M. Ishizuka, "Making topic specific report and multimodal presentation automatically by mining the web resources," in Proc. IEEE/WIC/ACM Int. Conf. Web Intell., 2006, pp. 240–246.

[15] M. Sravanthi, C. R. Chowdary, and P. S. Kumar, "SlidesGen: Automatic generation of presentation slides for a technical paper using summarization," in Proc. 22nd Int. FLAIRS Conf., 2009, pp. 284–289.

[16] M. Sravanthi, C. R. Chowdary, and P. S. Kumar, "QueSTS: A query specific text summarization approach," in Proc. 21st Int. FLAIRS Conf., 2008, pp. 219–224.

[17] H. P. Luhn, "The automatic creation of literature abstracts," IBM J. Res. Develop., vol. 2, pp. 159–165, 1958.

[18] P. B. Baxendale, "Machine-made index for technical literature: an experiment," IBM J. Res. Develop., vol. 2, no. 4, pp. 354–361, 1958.

[19] H. P. Edmundson, "New methods in automatic extracting," J. ACM, vol. 16, no. 2, pp. 264–285, 1969.

[20] A. Abu-Jbara and D. Radev, "Coherent citation-based summarization of scientific papers," in Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol.-Volume 1, 2011, pp. 500–509.

[21] V. Qazvinian, D. R. Radev, S. M. Mohammad, B. J. Dorr, D. M. Zajic, M. Whidby, and T. Moon, "Generating extractive summaries of scientific paradigms," J. Artif. Intell. Res., vol. 46, pp. 165–201, 2013.

[22] V. Qazvinian and D. R. Radev, "Identifying non-explicit citing sentences for citation-based summarization," in Proc. 48th Annu. Meeting Assoc. Comput. Linguistics, Jul. 2010, pp. 555–564.

[23] V. Qazvinian and D. R. Radev, "Scientific paper summarization using citation summary networks," in Proc. 22nd Int. Conf. Comput. Linguistics-Volume 1, Aug. 2008, pp. 689–696.

[24] Q. Mei and C. Zhai, "Generating impact-based summaries for scientific literature," in Proc. ACL, vol. 8, pp. 816–824, 2008.

[25] M. A. Whidby, "Citation handling: Processing citation texts in scientific documents," Doctoral dissertation, Dept. Comput. Sci., Univ. Maryland, College Park, MD, USA, 2012.

[26] R. Jha, A. Abu-Jbara, and D. Radev, "A system for summarizing scientific topics starting from keywords," ACM Comput. Surv., vol. 40, no. 3, p. 8, 2013.

[27] S. Mohammad, B. Dorr, M. Egan, A. Hassan, P. Muthukrishan, V. Qazvinian, D. Radev, and D. Zajic, "Using citations to generate surveys of scientific paradigms," in Proc. Human Lang. Technol.: The Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2009, pp. 584–592.

[28] P. Nakov, A. Schwartz, and M. Hearst, "Citation sentences for semantic analysis of bioscience text," in Proc. SIGIR'04 Workshop Search Discovery Bioinformatics, 2004, pp. 81–88.

[29] N. Agarwal, K. Gvr, R. S. Reddy, and C. P. Rose, "Towards multi-document summarization of scientific articles: Making interesting comparisons with SciSumm," in Proc. Workshop Autom. Summarization Different Genres, Media, Lang., 2011, pp. 8–15.

[30] O. Yeloglu, M. Evangelos, and Z.-H. Nur, "Multi-document summarization of scientific corpora," in Proc. ACM Symp. Appl. Comput., 2011, pp. 252–258.

[31] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," in Proc. ACL Workshop Intell. Scalable Text Summarization, 1997, vol. 17, no. 1, pp. 10–17.

[32] D. Marcu, "From discourse structures to text summaries," in Proc. ACL Workshop Intell. Scalable Text Summarization., 1997, vol. 97, pp. 82–88.

[33] I. Mani and E. Bloedorn, "Summarizing similarities and differences among related documents," Inf. Retrieval, vol. 1, no. 1, 2000, pp. 35–67.

[34] G. Erkan and D. R. Radev, "LexPageRank: Prestige in multi-document text summarization," in Proc. EMNLP, 2004, pp. 365–371.

[35] R. Mihalcea and P. Tarau, "A language independent algorithm for single and multiple document summarization," in Proc. IJCNLP, 2005, pp. 19–24.

[36] M. J. Conroy and D. P. O'leary, "Text summarization via hidden Markov models," in Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2001, pp. 406–407.

[37] D. Shen, J. T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in Proc. 20th Int. Joint Conf. Artif. Intell., 2007, vol. 7, pp. 2862–2867.

[38] Y. Ouyang, S. Li, and W. Li, "Developing learning strategies for topic-based summarization," Proc. 16th ACM Conf. Conf. Inf. Knowl. Manage., Nov. 2007, pp. 79–86.

[39] D. Galanis and P. Malakasiotis, "AUEB at TAC 2008," in Proc. Text Anal. Conf., 2008.

[40] R. McDonald, "A study of global inference algorithms in multi-document summarization," in Proc. Eur. Conf. Inf. Retrieval, 2007, pp. 557–564.

[41] D. Gillick, B. Favre, and D. Hakkani-Tur, "The ICSI summarization system at TAC 2008," in Proc. Text Anal. Conf., 2008.

[42] D. Gillick and B. Favre, "A scalable global model for summarization," in Proc. Workshop Integer Linear Program. Nat. Lang. Process., 2009, pp. 10–18.

[43] T. Berg-Kirkpatrick, D. Gillick, and D. Klein, "Jointly learning to extract and compress," in Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol., 2011, pp. 481–490.

[44] K. Woodsend and M. Lapata, "Multiple aspect summarization using integer linear programming," in Proc. Joint Conf. Empirical Methods Nat. Lang. Process. Comput. Nat. Lang. Learn., 2012, pp. 233–243.

[45] D. Galanis, G. Lampouras, and I. Androutsopoulos, "Extractive multi-document summarization with integer linear programming and support vector regression," in Proc. COLING, 2012, pp. 911–926.

[46] V. Vapnik, Statistical Learning Theory. Hoboken, NJ, USA: Wiley, 1998.

[47] C. C. Chang and C. J. Lin, (2001), LIBSVM: A library for support vector machines, [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm

[48] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang, "MEAD - A platform for multidocument multilingual text summarization," in Proc. 4th Int. Conf. Lang. Resources Eval., 2004, pp. 1–4.

[49] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford Digital Libraries, Stanford, CA, USA, Tech. Report: SIDL-WP-1999-0120, 1999.

[50] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," Phys. Rev. E, vol. 70, no. 6, p. 066111, 2004.

[51] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," J. Artif. Intell. Res., vol. 22, no. 1, pp. 457–479, 2004.

[52] C. Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in Proc. Workshop Text Summarization Branches Out, Post-Conf. Workshop ACL, 2004, pp. 25–26.
[53] A. Nenkova and R. J. Passonneau, “Evaluating content selection in summarization: The pyramid method,” in HLT-NAACL, vol. 4, pp. 145–152, May 2004.
[54] S. Modgil, N. Faci, F. Meneguzzi, N. Oren, S. Miles, and M. Luck, “A framework for monitoring agent-based normative systems,” in Proc. 8th Int. Conf. Auton. Agents Multiagent Syst., 2009, pp. 153–160.