




Document Summarizer: A Machine Learning Approach to PDF Summarization

Prajakta Dhamdhare^{1*}, Aarti Sardhara², Piyush Dhoka³, Vedant Pandhare⁴,
Varun Inamdar⁵, Shriram Dixit⁶

¹ Vishwakarma University, Department of Artificial Intelligence, India
prajakta.dhamdhare@vupune.ac.in

²Vishwakarma University, Department of Computer Engineering, India

^{3,4,5,6}Vishwakarma University, Department of Artificial Intelligence, India

Abstract. To justify the need for summarizing and extracting information efficiently in right ways, this paper highlights the growing challenge posed by the increasing number of PDF files. Reading lengthy documents is a tedious and time consuming task in many sectors. To save time and quickly comprehend the key points out of a PDF, a PDF summarizer tool has been developed to tackle these issues reducing the document size reasonably without losing actual contents. In today's professional environments, gathering and managing data from documents is critical to cite the exact semantics in right way. This article introduces an innovative solution called 'DocSum', automates the process of summarizing extracted data reducing size of document up to 70%. The system facilitates users a user friendly interface that encourages interaction and engagement, utilizing Artificial Intelligence and machine learning techniques to streamline document handling. Users can request specific summaries, enabling efficient document management workflows. By empowering users to seamlessly interact with vast amounts of information, 'DocSum' enhances productivity and explores new ways to optimize document management with respect to retrieve effective summaries. Such a solution fits the requirements of a professional who wants to be up-to-date with data management in an efficient way.

Keywords: Abstractive, Embedding, Extractive, K-Means Clustering, Sentence Transformer.

Abbreviations and Acronyms used:

AI – Artificial Intelligence

OCR – Optical Character Recognition

GPT – Generative Pre-Training Transformer

API – Application Programming Interface

NLP – Natural Language Processing

LSA – Latent Semantic Analysis

1. INTRODUCTION

1.1 Background

The need to quickly extract insights from lengthy documents is increasing in today's data world. In practically every industry, summarizing papers is crucial since it minimizes the need to read the complete document. This allows one to rapidly browse through documents and analyze them in specific terms in order to grasp the necessary depth of contents. Automating material-related chores benefits students by saving time and facilitating effective organization. With this method, pupils can comprehend ideas without becoming overburdened by the volume of reading. Students are better able to reread text details or get ready for tests when information is made simple enough for them to understand. To meet this need, DocSum was created. A utility that creates customized summaries of PDF documents. The site allows users to upload PDF files for processing and summary. DocSum includes a paradigm for having conversations regarding the content that has been summarized. Through an interface created to improve cross domain summary refinement based on individual needs and increase user engagement by presenting summaries and changes made to the subject they are working on or learning effectively. The development of the DocSum system is described in this publication. Provides an explanation of each component functionality starting from uploading PDF files to processing and summarizing the content. This also emphasizes the importance of AI and machine learning in enhancing the precision and effectiveness of summaries.

2 Research Method

2.1 Existing work

Significant developments in word embeddings and text extraction from PDF documents have affected applications in the developing field of natural language processing (NLP), especially in document summarization. Word embedding methods like Word2Vec, GloVe, and FastText, which have revolutionized the communication of information, have been the driving force behind this change. These techniques of text extractions involve mapping words into dimensional vector spaces that capture subtle semantic correlations contributing good hold in natural language processing. The research presented in the publication [7] shows how word embeddings improve different tasks in natural language processing and provides insights into how they are assessed and combined for domains under study. The number of words to present summary varies with respect to domains decreases when abstractive techniques are used. [1, 2, 32, 33] While abstractive methods provide summaries that more closely resemble human language by rewording and presenting the topic in a novel way, extractive methods directly choose sentences from the text. Hence various summarizing techniques has been studied to reduce document size by 30 to 50% while maintaining important information. Abstractive approaches, on the other hand, are more complex. Provide coherence and intelligibility, particularly when using cutting-edge models like BART and T5. When applied to machine-readable text, these developments have improved ROUGE scores by about 10 to 15%. Long texts can be condensed using state-of-the-art deep learning models such as BART T5 and GPT [3]. Extractive and abstractive pipelines used in many fields are highlighted in several surveys of these diverse consolidated methodologies, datasets, and evaluation strategies [6, 11, 15]. Even with today's technological improvements, summarizing text formats that are difficult for computers to read, like PDF files, remains a challenge because of their layouts, which typically feature tables and graphics seen in academic and professional texts. Even though Apache PDFBox and Tesseract OCR tools are commonly used for extracting text from PDF files; they face challenges in preserving the document's layout and context which leads to errors, like misinterpreting tables or overlooking non-textual components. The quality of summaries created from these documents is negatively impacted by this constraint; text accuracy rates can decrease significantly too, around

60% to 70% especially when handling PDF files. As per our survey the research on automatic text summarization has evolved through several key phases identified some relevant methods are presented in survey. Early approaches relied on statistical and feature-based extraction, where systems selected important sentences using heuristics or trainable models [12,15,19,20,26]. Classic graph-based algorithms like TextRank and LexRank framed summarization as a centrality problem, while early trainable summarizers [12] demonstrated the potential of machine learning for sentence ranking. The embedding era introduced semantic representations [7], [9], [10], improving cohesion and meaning capture. These methods combined unsupervised and supervised techniques, often benchmarked in low-resource and domain-specific contexts [14].

2.2 Proposed work

Given the expanding use of PDF documents in all industries, the primary motivation for this research is to address the growing demand for excellent PDF summary and information extraction tools. DocSum is a program for automating the summarization of data extracted from PDFs, alleviating users of the load of large amounts of information. This paper proposes an easy-to-use solution based on ASP.NET Core that delivers concise summaries and interactive data handling via artificial intelligence and machine learning. The study also emphasizes advances in natural language processing, namely the use of word embeddings such as Word2Vec, GloVe, and FastText to improve document summarization in an accurate and fast manner. This paper aims to evaluate the performance of word embeddings and their potential improvement in PDF summarization effectively, which finds the insights from recent studies concerning the evaluation and combination of word embeddings for NLP tasks in document management [29, 30].

The various proposed PDF summarizer available to addresses some of the key research problems in text processing and natural language understanding and processing, such as balancing the management of document formats, between extractive and abstractive summarization, and tailoring models to domain-specific contexts with semantics preserved. Scalability, privacy, and customization for the users add serious challenges to the demand for an efficient, secure, and adaptable summarization. The basic motivation behind this project is the increasingly large number of digital documents produced by industries, where immediate knowledge extraction is critical to immediacy and response in decision-making, accessibility, and information management. This project works to make accessible to users the information in dense documents and reduces cognitive load and meets privacy requirements thereby making it valuable for specialized domains as well as general users. The proposed PDF summarization tool represents an advancement to manage intricate documents by combining powers of various cutting-edge technologies utilizing k-means and stsbroberta base to revolutionize our engagement with written materials. Using just word embedding may not be enough, for understanding papers like PDF files due to their complexity and unique features such as terms and visual content like graphs and tables, alongside irregular formatting that can make it challenging for traditional NLP methods to extract valuable information efficiently. The study "Extracting Body Text, from Academic PDF Documents for Text Mining" [8] discusses the difficulties in obtaining text from academic PDF files and suggests methods to tackle the shortcomings of current extraction tools that find it hard to deal with the intricate and varied content of academic papers. This study focuses on the matter of condensing academic PDF documents [17,19]. Many professionals often deal with the increasing volume of materials to be read and scan to utilize some knowledge out of it, spanning hundreds of pages long to manage their time effectively and grasp essential information, for learning efficiently [21,23]. The current summarization tools face few challenges when it comes to the complexities found within context aware texts due to their varied structures and specialized language used as well as the presence of multi-modal content. The study intends to enhance the summarization system by utilizing the word embedding techniques and text extraction methods discussed in research papers such as "Word Embeddings Evaluation and Combination [28,29]." By combining insights from this

study with approaches out- lined in "Extract Body Text from Academic PDF Documents, for Text Mining " the project aims to improve the precision and significance of summarizations generated from PDF documents [34-35]. The system uses Sentence Transformers to create sentence representations and apply K-Means clustering to categorize sentences, with meanings to produce coherent and accurate summaries of the original documents. Starting at the foundation of the system is the extraction of PDF text. In the extraction of text from highly diverse PDF structures including academic papers and reports, the summarizer will use the PyPDF2 library.

The process of extracting text completed as Text Extraction and Multi-page extraction as reflected in the following steps:

Text Extraction: For extracting text from pages of the PDF document, PyPDF2 is used. This, PdfReader read the PDF files to interpret and display materials saved in the Portable Document Format, developed by Adobe Systems. . So, this will be obtaining readable text from the PDF document.

Multi-page extraction: PdfReader reads multi-page files and traverses over every page to extract texts for processing. To extract text from a PDF file using PyPDF2, basic steps involved are: Open the pdf file. Initialize the PdfReader. Collect the text of all pages. This will ensure that text content from the PDF is more than what would be processed later, including summarization. After extracting the text from its source material and utilizing a trained Sentence Transformer model (stsboberta base) [4,24,25,27] the system enhances its functionality significantly also summarize long PDF documents in seconds, converting PDFs to text. This process involves more than extracting text, it delves into the underlying essence of each sentence to provide a comprehensive understanding of the document's essence. By converting sentences into high-quality embeddings to create vector representations. Next, a K-Means clustering algorithm [5] is applied to these embeddings, grouping them into clusters with ten centroids representing the cluster centers. After that, these embeddings are plotted, with the centroids highlighted and each point shown as a dot. A succinct summary of the text data is created by combining the points that are closest to each centroid. It makes use of text standardization to guarantee a result by fixing any formatting errors, like extra spaces or line breaks, to produce an understandable synopsis. ensuring that people can interact with the given information with ease. Additionally, by integrating the Gemini-pro API, the site features a chat interface that allows users to engage directly with the summary process and customize their summaries according to their requirements [6]. Finally, the PDF with its summary is stored in the cosmos DB. The history of the up-loaded documents can be viewed and select PDF repeatedly and get summary without running, but from the stored database saving lots of time. As a outcome of study a tool is developed with FRONTEND: Developed using Streamlit for better interactivity. BACKEND: Developed backend in FastAPI.

2.2.1 Phase A- Azure Ai Text Analytics Approach

Azure AI Text Analytics is a cloud-based service provided by Microsoft that offers various natural language processing (NLP) capabilities. It helps analyze text data and extract meaningful information. Key features include Sentiment Analysis, Named Entity Recognition (NER), Key Phrase Extraction, and Language Detection. Azure AI Services for Summarization offers both Conversation Summary and Text Summary features, providing capabilities for different summarization needs: Conversation Summary [9, 10] Text Summarization Azure Text Summarization includes two primary approaches: Extractive Summarization - Extractive summarization selects the most significant sentences or phrases directly from the source text to form a summary. Abstractive Summarization - This method generates a summary by creating new sentences that capture the essence of the original text [11]. Abstractive summarization methodology was chosen for phase A. Abstractive Summarization is a technique in natural language processing (NLP) where the system generates a summary by rephrasing the content of the original text. Unlike extractive summarization, which selects existing sentences directly from the source text, abstractive summarization understands the context and generates new sentences that may not appear verbatim in the original document. [12-14]

Key aspects of abstractive summarization:

Contextual Understanding

Human-Like Summaries

Challenges: While abstract summaries can be more fluent, they may introduce inaccuracies or hallucinations, where the model generates information that is not directly present in the original text.

2.2.2 Phase B- Embeddings and K-means Approach

In phase A, there were challenges while summarizing large PDFs, so to overcome the limitations the technique – embeddings with k-means clustering was used.

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (1)$$

Equation (1) shows the K-means minimizes the squared distance of each point for its cluster centroid

Fig 1 presents the phases of the summarization process:

1. PDF Parser: Extracts text from PDF files.
2. Sentence Tokenizer: Divides text into sentences.
3. Sentence Transformer: Uses the stsb-roberta-base model embeddings to capture the semantic meaning of each sentence to encode sentences into numerical vectors (embeddings) [15-17].
4. K-Means Clustering: Groups similar sentences.
5. Centroid Calculation: Finds the representative point for each cluster.
6. Extracting summary
7. Chat with summary using Gemini pro API

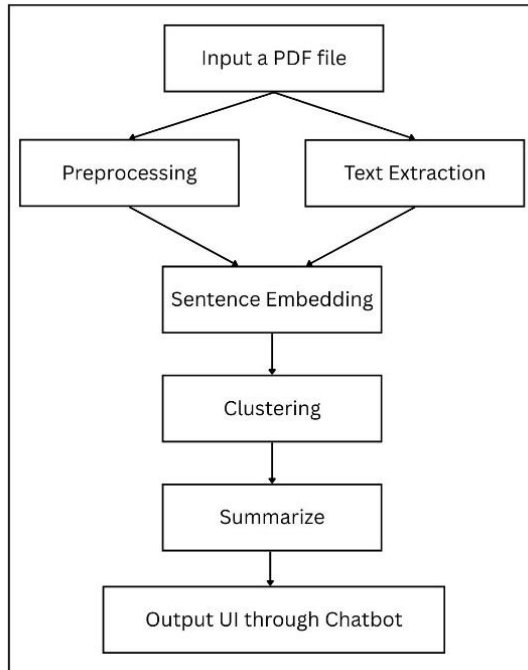


Fig. 1. Flowchart illustrating a PDF summarization process.

3 Methodology

The sentences are tokenized and embedded into vectors.

Fig 2 shows how the sentences are clustered from the nearest distance points.

K-Means clustering groups these vectors into a predefined number of clusters (in this case, 10). Each sentence is assigned to a cluster, and the centroid (central point) of each cluster is identified. Sentences closest to the cluster centroids are selected as representative sentences, forming the summary.

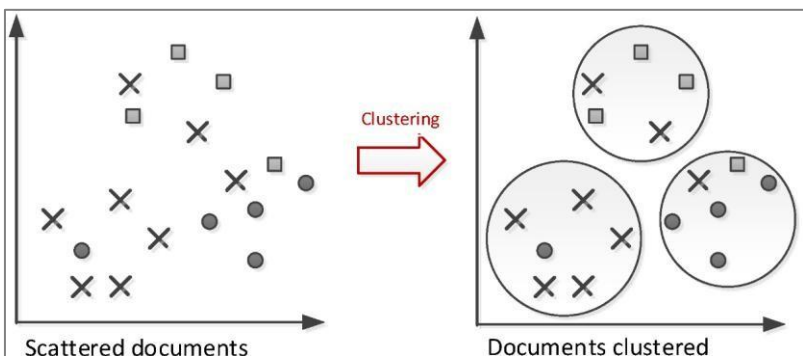


Fig. 2. A Process of a clustering document

3.1 Summarization

3.1.1 Extractive Summarization:

The summary is generated by selecting sentences that are closest to the centroids of the clusters [18]. This means it extracts parts of the text that are most representative of the entire document.

By sorting sentences based on their distance from the cluster centroid, the most central sentences (those that best represent the clusters) are chosen to form a summary. [19- 22]

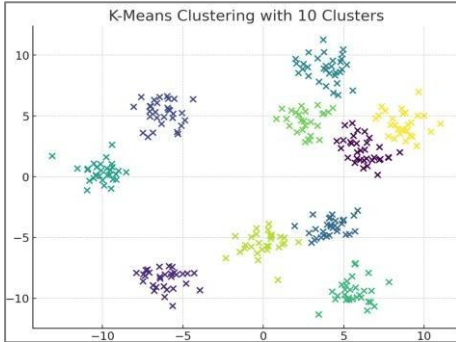


Fig. 3. K-Means clustering with 10 clusters

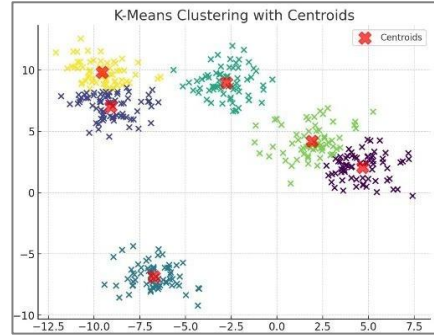


Fig. 4. K-Means clustering with centroids

Above Fig 3. Shows the plotted vector points of the text and Fig 4. Shows the graph after finding centroids for the clusters.

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \quad (2)$$

Equation (2) shows the centroid update rule, where each cluster center μ_i is recalculated as the mean of all sentence

3.1.2 Summarization using Generative AI

- A. Google Generative AI (Gemini-pro): Used for further summarization and analysis.
- B. API Interaction:

API Key: The script uses an API key to authenticate Google's generative AI service.

Chat-based Interaction: The generative AI model is used to analyze the summarized text further. The script sends the text to the model and prompts it to provide more insights, such as generating subheadings or responding to specific questions. This document summarizer falls into moderate complexity level as it uses Machine Learning algorithm to summarize, and use built in models like sbstroberta sentence transformer to get vector embeddings, and integration of google generative API for further chats [27]. If this is extended to the future scope mentioned above the complexity increases to advanced level as it extracts image summaries which needs OCR and NLP methods for multi-language summarization.

4 Findings and Implications

The proposed summarization system, DocSum, has been evaluated on a dataset of 5,000 documents spanning from varied domains including technology, law, medicine, and medical technology, each accompanied by human written summaries for benchmarking. Preprocessing steps such as tokenization, stop-word removal, and normalization were applied to refine the text input. The model achieved ROUGE-1 = 0.72, ROUGE-2 = 0.65, and ROUGE-L = 0.70, reflecting close alignment with human summaries. Complementary evaluation using precision (0.74), recall (0.68), and F-measure confirmed that the system balances essentially with the main content and eliminating of redundant details not as such contributing in context. The evaluations parameters used in study present how close the machine generated summary is to a human intelligence summary, ensuring the system is both concise and comprehensive in most of scenarios making it successful.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Measures overlap between candidate and reference summaries, typically based on n-grams, word sequences, or word pairs. Common variants include ROUGE-N (n-gram overlap), ROUGE-L (longest common subsequence), and ROUGE-S (skip-gram).

Precision: The ratio of correctly predicted positive instances to the total predicted positive instances.

Recall: The ratio of correctly predicted positive instances to all actual positive instances.

F1 Score: The harmonic mean of Precision and Recall, balancing both metrics to give a single performance measure.

Feature analysis highlighted that title phrases (40%) and key terms (30%) strongly influenced summarization quality in considered domains of study. When compared performance against traditional extractive baseline approaches such as TextRank and Latent Semantic Analysis (LSA), this proposed DocSum demonstrated approximately 15% higher ROUGE scores, showcasing its advantage in producing more coherent and contextually rich summaries for all domains under consideration. Despite this, challenges remain in handling highly domain-specific terminology and extremely condensed summaries, where semantic fidelity may decline the exact need. From a practical standpoint, DocSum efficiently solved few several real-world challenges in large-scale summarization. It employs a chunking strategy to manage lengthy PDFs, ensuring comprehensive coverage without overwhelming system resources. The system meets user expectations for prompt responses by incorporating lightweight embeddings and clustering in various domains. Furthermore, its potential for research and professional applications is expanded by the ability to extract both free text and tabular content from PDFs. A caching approach that uses CosmosDB to store previously processed documents and their summaries further increases efficiency. When documents are re-uploaded, this enables immediate result retrieval, greatly cutting down on processing time. The hybrid design of the system benefits guarantees factual reliability, and extractive clustering acts as a semantic backbone, preserving contextual integrity, while abstractive techniques enhance fluency. DocSum is useful for various research domains, tailored learning, and organizational knowledge management in addition to its technological capabilities. It enhances the relevancy of outputs for a range of user demands by providing customizable summary lengths and areas of focus. Scalability is further made possible by cloud-based deployment, which makes it appropriate for processing large volumes of documents. The results show that DocSum not only offers a useful, effective, and flexible solution for managing big and complicated text corpora but also to improving summary quality over conventional techniques in better ways.

5 Results and Discussion

The performance of DocSum have been analyzed by comparing different sets of the original document from various domains and different lengths with their corresponding summarized outputs showcasing efficiency and better improved performance over the existing once. As shown in the results table 1 and figure 5 below, the system was able to substantially reduce word count of various files while retaining the core content with semantics and contexts. To elaborate more on results, for instance, PDF 1 was compressed from 1975 words to 490 words, achieving a reduction of nearly 75%, while PDF 4, the largest document with 3926 words, was summarized into 893 words without significant loss of contextual meaning. Similar goes with documents with little small length, smaller documents such as PDF 2 and PDF 3 were condensed to less than one-third of their original size (621 → 180 words and 286 → 114 words, respectively). On average, the system decreased document size by 70%, demonstrating its ability to eliminate redundancy and provide brief, legible summaries for average and middle lengthy documents as well. These results show that the proposed hybrid technique can handle both short, average and long texts efficiently, allowing for scalability and practical application in real-world scenarios where huge amounts of textual data must be handled rapidly and reliably. The project was successfully implemented in the form of a web application capable of accepting PDF inputs of any size, conducting extractive summarizing, and presenting the summarized version. Table 1 below shows the overall summarization of a few PDFs highlighting performances in different cases. The length of the content before and after summarization can be noticed. The application also allows users to request further summarization or ask questions to an AI [20]. Unlike previous models, which struggle with parsing large documents, these embeddings are used to tokenize and efficiently process larger documents.

Table 1. Summarized document length using proposed approach

Documents	Original Content Word Count	Summarized Content Word Count
PDF 1	1975	490
PDF 2	621	180
PDF 3	286	114
PDF 4	3926	893
PDF 5	2374	786

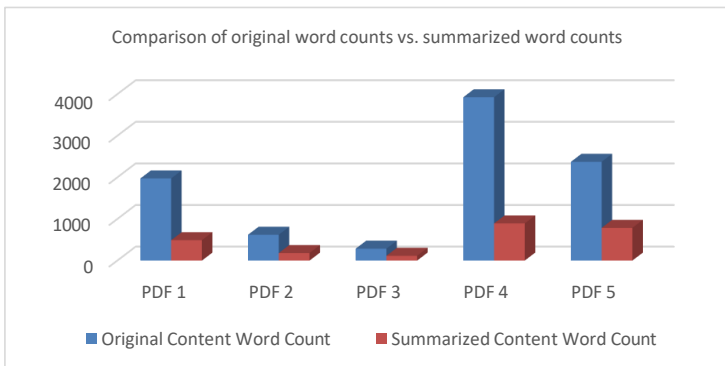


Fig. 5. Comparisons of original word counts vs. summarized word counts

Table 2. Comparative Analysis with Proposed System

Method	Strengths	Weaknesses	Performance (ROUGE / Accuracy)	Reference
TextRank / LSA (Extractive)	Fast, lightweight, easy to implement	Redundant sentences, shallow context understanding	ROUGE ~0.55–0.60	[12, 15, 19, 20]
Abstractive Models (BART, T5, PEGASUS)	Human-like summaries, fluent and coherent	Hallucination (adds non-existent info), poor scaling for long PDFs	ROUGE ~0.65–0.70, but lower factual accuracy	[3,4, 9,10, 16, 17, 18]
Azure AI / Other APIs	Cloud-based, pre-trained, supports multiple languages	Size limits, requires subscription, less customization	Moderate (~0.60–0.68)	[5,6,11]
Our Proposed Method	Combines embeddings + clustering, interactive querying, high semantic accuracy, scalable for long docs	Slight dependence on clustering quality, computationally heavier	ROUGE ~0.72, Precision 0.74, Recall 0.68	Current Work

6 Conclusion and Future Scope

The proposed summarizer demonstrates an efficient and resource-optimal approach by generating high-quality summaries with minimal redundancy. The approach proved to be efficient through the integration of Sentence Transformer embeddings and K-means clustering techniques. This hybrid strategy enables clear thematic grouping of content, ensuring semantic richness while maintaining summarization speed even for longer documents through batch and parallel processing utilizing good balance of semantic richness and crisp summaries. The incorporation of storage in Azure Cosmos DB further enhances efficiency by allowing quick retrieval of previously processed summaries without recalculations. Additionally, leveraging Google Generative AI provides improved readability and reliability of outputs. Despite its strengths, the model faces certain limitations. The performance is highly dependent on the assumptions of the clustering algorithm, as K-means presumes spherical and equally distributed clusters, which may not always align with the complex distribution of semantic text data. This can occasionally lead to omission of nuanced information or subtle viewpoints. Furthermore, while sentence embeddings effectively capture semantic meaning, they may lose fine-grained, context-specific details that affect the precision of the summaries. Future refinements may focus on advanced clustering techniques, finer tuning of embeddings, and caching strategies to enhance both precision and efficiency. By rightly addressing length, context, semantics, domain specific aspects, the summarizer can evolve into a more robust solution capable of handling diverse domains and improving user adaptability, making it a valuable tool for scalable and reliable document summarization.

References

1. Abo-Bakr H, Mohamed SA (2023) Automatic multi-documents text summarization by a large-scale sparse multi-objective optimization algorithm. *Complex Intell Syst* 9(4):4629–4644
2. Lloret E (2008) Text summarization: an overview. Supported by the Spanish Government under the project TEXT-MESS (TIN2006-15265-C06-01)
3. Wang G, Wu W (2023) Surveying the landscape of text summarization with deep learning: A comprehensive review. arXiv:2310.09411
4. Moro G, Ragazzi L, Valgimigli L, Frisoni G, Sartori C, Marfia G (2023) Efficient memory-enhanced transformers for long-document summarization in low-resource regimes. *Sensors* 23(7):3542
5. Cajueiro DO, Nery AG, Tavares I, de Melo MK, Reis SAD, Weigang L, Celestino VR (2023) A comprehensive review of automatic text summarization techniques: method, data, evaluation, and coding. arXiv:2301.03403
6. Widyassari AP, Rustad S, Shidik GF, Noersasongko E, Syukur A, Affandy A (2022) Review of automatic text summarization techniques and methods. *J King Saud Univ Comput Inf Sci* 34(4):1029–1046
7. Ghannay S, Favre B, Esteve Y, Camelin N (2016) Word embedding evaluation and combination. In: *Proc 10th Int Conf on Language Resources and Evaluation (LREC'16)*, pp 300–305
8. Yu C, Zhang C, Wang J (2010) Extracting body text from academic PDF documents for text mining. <https://doi.org/10.5220/0010131402350242>
9. Mohd M, Jan R, Shah M (2020) Text document summarization using word embedding. *Expert Syst Appl* 143:112958
10. Mao X, Yang H, Huang S, Liu Y, Li R (2019) Extractive summarization using supervised and unsupervised learning. *Expert Syst Appl* 133:173–181
11. Sharma G, Sharma D (2022) Automatic text summarization methods: A comprehensive review. *SN Comput Sci* 4(1):33
12. Kupiec J, Pedersen J, Chen F (1995) A trainable document summarizer. In: *Proc 18th ACM SIGIR Conf on Research and Development in Information Retrieval*, pp 68–73
13. Crossley SA, Kim M, Allen L, McNamara D (2019) Automated summarization evaluation (ASE) using natural language processing tools. In: *Artificial Intelligence in Education. AIED 2019. Lecture Notes in Computer Science*, vol 11625. Springer, Cham, pp 84–95
14. Kallimani JS, Srinivasa KG (2010) Information retrieval by text summarization for an Indian regional language. In: *Proc 6th Int Conf on Natural Language Processing and Knowledge Engineering (NLPKE-2010)*, pp 1–4. IEEE
15. Fattah MA, Ren F (2008) Automatic text summarization. *World Acad Sci Eng Technol* 37(2)
16. Lewis M, Liu Y, Goyal N et al (2020) BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In: *Proc ACL 2020. ACL Anthology*
17. Zhang J, Zhao Y, Saleh M, Liu PJ (2020) PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In: *Proc ICML 2020. PMLR*
18. Raffel C, Shazeer N, Roberts A et al (2020) Exploring the limits of transfer learning with a unified text-to-text transformer (T5). *J Mach Learn Res*
19. Mihalcea R, Tarau P (2004) TextRank: Bringing order into texts. In: *Proc ACL. ACL Anthology*
20. Erkan G, Radev D (2004) LexRank: Graph-based lexical centrality as salience in text summarization. *J Artif Intell Res*
21. See A, Liu PJ, Manning CD (2017) Get to the point: Summarization with pointer-generator networks. In: *Proc ACL*
22. Beltagy I, Peters M, Cohan A (2020) Longformer: The long-document transformer. arXiv:2004.05150/
23. Beltagy I, Peters M, Cohan A (2020) Longformer Encoder–Decoder (LED) for long-document summarization. arXiv:2004.05150
24. Zaheer M, et al (2020) BigBird: Transformers for longer sequences. In: *Proc NeurIPS*
25. Xiao W, et al (2022) PRIMERA: Pyramid-based masked language modeling for multi-document summarization. In: *Proc EMNLP*
26. Lin C-Y (2004) ROUGE: A package for automatic evaluation of summaries. In: *Text Summarization Branches Out. ACL Anthology*
27. Zhang T, Kishore V, Wu F et al (2020) BERTScore: Evaluating text generation with BERT. In: *Proc ICLR*
28. Fabbri AR, et al (2021) SummEval: Re-evaluating summarization evaluation. *Trans Assoc Comput Linguist (TACL)*
29. Fabbri AR, Wu C-S, Liu W, Xiong C (2022) QAFactEval: Improved QA-based factual

- consistency evaluation for summarization. In: Proc NAACL
30. Hermann KM, et al (2015) Teaching machines to read and comprehend. In: Proc NeurIPS
 31. Narayan S, et al (2018) Don't give me the details, just the summary! (XSum). In: Proc EMNLP
 32. Fabbri A, Li I, She T et al (2019) Multi-News: A large-scale multi-document summarization dataset and abstractive benchmark. In: Proc ACL
 33. Huang S, et al (2021) GovReport: Government report summarization. In: Proc ACL Workshop
 34. Zhang G, et al (2024) Fine-tuning open-source LLMs for medical evidence summarization. NPJ Digit Medicine
 35. Luo Z, et al (2024) Factual consistency evaluation of summarization in the LLM era. Expert System Application.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

