International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Summarize-AI

Pallavi Maddula[1], Khaizar Kanchwala[2], Khaja Boqthiyar[3], P Nageswara Rao[4]

[1,2,3]Student, Department of Computer Science & Engineering(AIML), Neil Gogte Institute of Technology, Kachavanisingaram, Telangana, India
[4]Assistant Professor, Department of Computer Science and Engineering, Neil Gogte Institute of Technology, Kachavanisingaram, Telangana, India

Abstract
Text summarization is a system which generates a shorter and a precise form of one or further textbook documents. Automatic textbook summarization plays an essential part in chancing information from large textbook corpus or an internet. What had actually started as a single document Text Summarization has now evolved and developed into generating multi-document summarization. There are a number of approaches to multi document summarization similar as Graph, Cluster, Term- frequence, idle Semantic Analysis (LSA) grounded etc. In this paper we've started with preface of multi-document summarization and also have further bandied comparison and analysis of colorful approaches which comes under the multi-document summarization. The paper also contains details about the benefits and problems in the being styles. This would especially be helpful for experimenters working in this field of textbook data mining. By using this data, experimenters can make new or mixed grounded approaches for multi document summarization.

Keywords: Text summarization, cluster, multidocument summarization, graph, LSA, TermFrequency Based.

1. Introduction

For reacquiring information, People extensively use internet similar as Google, Yahoo, Bing and so on. Since quantum of material on the internet is growing fleetly, for druggies it is not easy to find applicable and applicable information as per the demand. Once a stoner sends a query on a hunt machine for data or information also the response is utmost of the times thousands of documents and the stoner has to face the tedious task of chancing the applicable information from this ocean of answer. This problem is called as "Data Overloading" [1]. Automatic textbook summarization is the summary of source of textbook in shorter interpretation, that retain the main point of the content and help the stoner to snappily understand large volume of information. A number of authors have proposed ways for automatic textbook summarization which can be astronomically classified as extractive summarization and abstractive summarization. In extractive summarization, it selects rulings that have the loftiest weightage in the recaptured document and put them together to induce a summary interpretation of original document without changing or altering the main textbook, where as in abstractive summary, the original textbook gets converted into another semantic form with the help of verbal styles to get a shorter summary of original document [3]. The primary thing of multiple- document summarization is to make summary

which has maximum content, lower spare data and maximum cohesiveness between rulings [2]. In another words, main rulings are uprooted from each document and also arere-arranged to get multi-documents summary. Multi-document summarization inflow is shown in Fig. 1 1Multi-Document Process Flow This check paper covers colorful aspects which are given below Several approaches of Graph, Cluster, Term frequence, and idle sematic analysis for multi-document summarization Issues and problems shown by different experimenters for enhancement in this area Evaluation criteria for comparing automatic summary and mortal summary We've in Section II of this paper, described affiliated work done on multi-document textbook summarization. In the Section III we've shown analysis and comparison of all styles with compass of enhancement, Section IV contains the evaluation criteria and Section V contains conclusion.

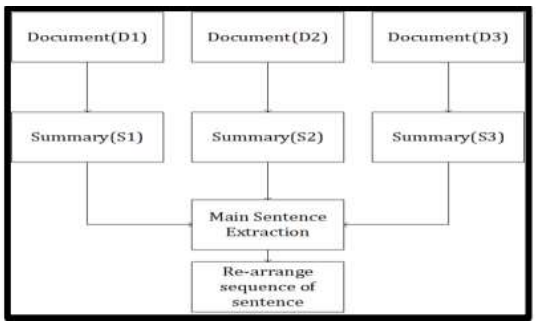

Fig 1: Multi-Document process flow

2. Analysis and Comparative Study of Various Methods

Category	Author, Year	Description	Benefits	Problems	Scope of Improvement
Grid Based Method	Rada Mihalcea, 2004 [4]	Builds summary using TextRank that selects top sentences	Considers text units for local information	Complex calculation of vertex score	Improve score calculation for better summary generation
	Shanmugasudaran Hariharan, 2009 [5]	Uses cumulative sum & degree of centrality for summarization	Works for both single & multi-document summarization; uses precision and recall	Only precision and recall used; lacks other evaluation formulas	Develop extra methods for better results
Graph Based Method	Tu-Anh Nguyen-Hoang, 2002 [6]	Uses preprocessing, graph construction, and MMR-based sentence ranking	Unsupervised method; no need for training data	Information loss may occur during graph construction	Optimize for reduced information loss
Cluster Based	Xiao-Chem Ma, 2009 [7]	Clusters and extracts	Uses MMR for effective	Only considers	Improve readability of

Method		summary using modified MMR	sentence extraction	query sentences	the generated summary
	Virendra, 2002 [8]	Merges single and multi-document summaries using syntactic and semantic similarity	Effective sentence clustering based on semantic features	Relies on word order for syntactic similarity; can use other structural comparison measures	Use alternative measures for syntactic similarity
Term Frequency Method	Salton, 2005 [9]	Summary generation using TF-IDF	Fast and easy summarization	No major drawbacks	Integrate other features to remove redundancy
	Jun'ichi Fukumoto, 2004 [10]	Multi-document summarization using single-document summarization	Categorizes documents as single-topic, multi-topic, or others	No major drawbacks	Improve result quality
LSA Based Method	Shuchu Xiong, 2004 [11]	LSA-based summarizer using SVD, MEAD, and MMR to select sentences based on prediction similarity	Applies SVD; uses centroid-based MEAD & MMR	Only LSA-based methods used	Combine with other techniques for improved summarization
	Josef Steinberger, 2004 [13]	Recalculates SVD of term-sentence matrix; uses topic similarity and term significance	Highlights topic similarity and term importance	No standard method used for summary evaluation	Develop a robust evaluation method

3. Methodology

This paper aims to design and implement a powerful yet resource-efficient system that allows users to summarize multiple documents and interact with their content using a chatbot—entirely on a local machine, without relying on cloud services. Built around Llama-2, an open-source large language model developed by Meta, the system provides users with the ability to upload multiple documents or text inputs, which are then processed to generate concise summaries. [14] Users can choose the summarization style

(creative, formal, detailed, or simple) and preferred language (English or Telugu), ensuring personalized and user-friendly output. In addition to summarization, the system integrates a document-aware chatbot that enables users to ask questions based on the content of the uploaded documents. This is achieved by embedding document chunks using MiniLM sentence transformers and storing them in a FAISS vector database for efficient semantic search. The LangChain framework is used to build both the summarization and QA logic, connecting the prompt, Llama-2 model, and relevant document context. To evaluate the relevance and accuracy of the generated summaries, the system includes a cosine similarity scoring mechanism, which compares the model-generated summary with a reference summary, if provided. This adds a layer of quality assurance to the process. The entire system is deployed using Streamlit, offering an intuitive, interactive, and easy-to-use web interface. It also supports text extraction from image-based documents using Tesseract OCR, making it versatile in handling different types of input. By leveraging the quantized 8-bit version of Llama-2 via Llama.cpp, [15] the system achieves significant performance gains, enabling smooth execution even on machines with limited computational power. This local-first approach ensures data privacy, lower latency, and broader accessibility. 9 Overall, the project provides a unified solution for document summarization, intelligent question-answering, and summary evaluation—tailored for educational, research, and professional use cases were understanding large volumes of information quickly and effectively is essential

3.1 System Architecture

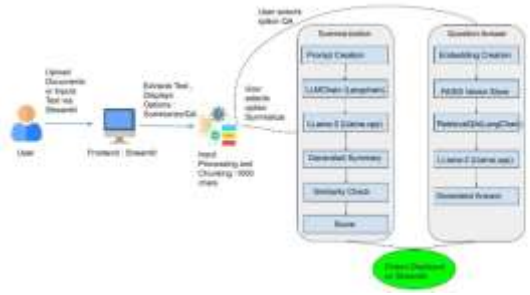

Fig 2: Architecture of the system.

The system architecture represents a document or text processing application that allows users to either summarize uploaded documents or perform question-answering tasks. The interaction begins with the user uploading documents or directly inputting text through a Streamlit-based frontend. Once the text is received, it undergoes text extraction, and the system displays two options for the user to choose from: Summarization or Question Answering (QA). Before further processing, the extracted text is chunked into blocks of 1000 characters to facilitate efficient handling, especially when dealing with large documents.
If the user selects the Summarization option, the system first creates a prompt tailored for summarization. This prompt is then processed through LLMChain, a module provided by LangChain, which helps manage the communication with the underlying large language model. The prompt is then passed to the Llama-2 model, deployed using llama.cpp for lightweight and efficient inference. The model generates a summary based on the input text. Following this, the generated summary undergoes a similarity check to evaluate how well it aligns with the original content, ensuring that the summary maintains the context and key points. Based on this comparison, a score is calculated to reflect the quality of the summarization.

Fig 3: Class connection of the system.

On the other hand, if the user selects the Question Answering (QA) option, the system first performs embedding creation by converting the input text into vector representations suitable for semantic search. These embeddings are stored in a FAISS vector store, a highly efficient library for similarity search. When a user poses a query, the system uses RetrievalQA from LangChain to retrieve the most relevant chunks from the FAISS vector database. These retrieved chunks are then sent to the Llama-2 model via llama.cpp, where the model generates a precise answer to the user's question based on the retrieved context.
Finally, whether the task was summarization or question answering, the output (summary, score, or answer) is displayed back to the user via the Streamlit interface, creating a seamless and interactive experience. This architecture effectively combines document processing, natural language generation, vector similarity search, and frontend display, all orchestrated to provide a user-friendly platform for text summarization and QA functionalities.

3.2 Implementation:

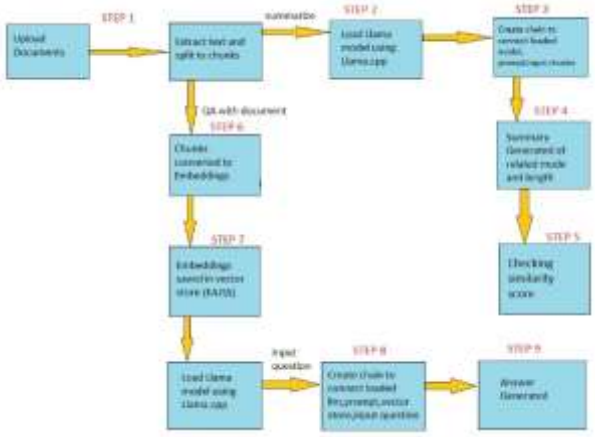

Fig 4: System Implementation flow.

Our solution is built on a modular architecture that uses the Llama-2 language model for multi-document summarisation, interactive document-based querying, and similarity-based quality evaluation—all of which can be deployed on local hardware.
The system is mostly written in Python, and Streamlit serves as the frontend, providing an intuitive, user-friendly interface for document submissions, summarisation, and chatbot interaction. To meet the project's objectives, the backend comprises numerous core modules as well as third-party libraries.

Summarisation and QA Modules:

The basic language model, Llama-2 (7B, 8-bit quantised GGUF), [16] is accessed through Llama.cpp, a lightweight C++ implementation that allows for efficient local inference without the need of GPUs.

LangChain orchestrates summarisation and question-answering duties by connecting the model to user prompts and chunked document information.

To efficiently analyse vast amounts of textual data, the CharacterTextSplitter function divides documents into digestible segments. These chunks are integrated with the MiniLM sentence transformer (using HuggingFace's all-MiniLM-L6-v2 model) to generate semantic vector representations. The embeddings are saved and retrieved using FAISS, an efficient vector search library.

Summarisation Pipeline:

When a user uploads a document (PDF, text, or photos processed with Tesseract OCR), the system extracts the text and allows the user to specify the summarisation language (English or Telugu), preferred tone (creative, formal, detailed, or simple), and word count. Summarisation prompts are created dynamically, resulting in succinct results according to user choices.

The system also allows audio output by translating summaries to speech via the gTTS library, which improves accessibility.

QA ChatBot Integration:

For interactive querying, the document-aware chatbot uses LangChain's Conversational Retrieval Chain. User enquiries are semantically matched to FAISS document embeddings, and appropriate context is supplied back to Llama-2 to provide correct, context-specific replies.

Summary of Quality Evaluation:

To assure the accuracy of the generated summaries, the system contains a cosine similarity score module. When a reference summary is supplied, the system estimates the cosine similarity of the model-generated summary to the reference text.[20] This transparent metric allows users to check the accuracy of generated summaries.

System Performance and Deployment:

To optimize resource usage and performance, the project uses an 8-bit quantized Llama-2 model and modular Python components.[21] The application is designed to be locally deployable on standard hardware, running efficiently on devices with at least 8 GB RAM and mid-range CPUs. All components—summarization, chatbot, similarity evaluation—are integrated into a single cohesive system, empowering users to navigate large textual data interactively and securely without reliance on cloud-based solutions.

4. Evaluation Measures

Evaluation of summary is typically constructed on readability and content of information. Primary purpose of text summarization is to find non redundant text that have contained significant information from the original corpus. There is no fixed parameter for text summarization on which we can rely for evaluation. There are two approaches for evaluation of summarization i.e. intrinsic and extrinsic. Intrinsic meth- od calculates the actual information of a summary, compares with human summary or with the full document source. In extrinsic methods evaluate the summary via task- based performance i.e. information retrieval-oriented tasks.

The Rouge toolkit can help us to check performance of the summary generated. Rouge is a software package which can be used to measure summary in period of number of words overlaps in machine generated summary and human reference summary [17]. In Rouge toolkit, as input, we can provide two types of summaries. Standard summary can be considered as location summary which we can compare our summary results and other that are generated via some methods. Rouge toolkit has five evaluation

metrics i.e. ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU based on word co-occurrence statistics [17].

There is another toolkit called MEAD which is a publicly open toolkit for multi- lingual summarization and evaluation. This toolkit implements several summarization algorithms i.e. position-based, centroid, TF-IDF, and query-based methods, etc. Methods for evaluating the quality of the summaries include co-selection (preci- sion/recall, kappa, and relative utility) and content-based measures (cosine, word overlap, bigram overlap).

5. Conclusion:

This literature survey paper contains various methods for multi-document text summarization. Several techniques have been explored for multi-document summarization such as Graph Based, Cluster Based, Term-Frequency Based and Latent Se- mantic Analysis(LSA) based.[18] Researchers can focus only on specific approaches from existing techniques and make an improvement in those approaches to generate new or hybrid approach for building better summaries which take less effort. We have com- pared in this paper, Graph, Cluster, Term-Frequency and LSA. New approach or hybrid approach can be developed with help of natural language processing approach and linguistic approach, which can help us to generate better summary for multi- document.

6. Future Scope

The Llama-2 based multi-document summarization and chatbot application lays a strong foundation, but there are several promising directions for future enhancement:

Domain-Specific Customization: Fine-tune Llama-2 models for specific industries like medical, legal, or academic research to provide more accurate and context-aware summarizations and answers.

Multilingual and Code-Mixed Language Support

Extend the current bilingual (English and Telugu) support to include more regional and international languages.

Handle code-mixed inputs (e.g., English + regional language) for better user interaction in diverse linguistic settings.

Voice-Based Interaction: Integrate speech-to-text and text-to-speech for voice-based document interaction and chatbot conversations, enabling better accessibility and usability.[19]

Improved Model Efficiency & Deployment: Explore further quantization techniques (e.g., 4-bit) and deployment options (e.g., ONNX, WebAssembly) for running on low-resource devices, including mobile or edge computing platforms.

Real-Time Summarization and QA: Enable real-time summarization and dynamic updates as documents are edited or updated, especially useful in collaborative environments.

Advanced Similarity Scoring: Incorporate multiple evaluation metrics (e.g., ROUGE, BERTScore) along with cosine similarity to improve the accuracy and reliability of summary evaluation.[20]

Knowledge Graph Integration: Link document content to structured knowledge graphs for more insightful question answering and contextual linking across documents.

Ethical and Privacy Considerations: Build features to detect and mitigate bias in summaries or answers, and implement privacy-preserving mechanisms for sensitive documents.

User Feedback Loop: Incorporate a feedback mechanism to allow users to rate summaries or chatbot answers, enabling continuous model fine-tuning and user-specific adaptation.[21]

References:

1. M.-y' Kan and 1. L. Klavans, "Using librarian techniques in automatic text summarization for information retrieval, " in Proceedings of the 2nd ACMlIEEE-CS joint conference on digital libraries, pp. 36-45, ACM, 2002
2. Y. K. Meena, A. Jain and D. Gopalani, "Survey on Graph and Cluster Based approaches in Multi-document Text Summarization," Recent Advances and Innovations in Engineering (ICRAIE), 2014, Jaipur, 2014, pp. 1-5. doi: 10.1109/ICRAIE.2014.6909126
3. M. Haque, S. Pervin, Z. Begum, et aI., "Literature review of automatic multiple documents text summarization, " International Journal of Innovation and Applied Studies, vol. 3, no. 1, pp. 121-129, 2003.
4. R. Mihalcea and P. Tarau, 'Textrank: Bringing order into texts, " in Proceedings of EMNLP, vol. 4, Barcelona, Spain, 2004.
5. S. Hariharan and R. Srinivasan, "Studies on graph based approaches for single and multi- document summarizations, " Int. 1. Comput. Theory Eng, vol. 1, pp. 1793-8201, 2009
6. T.-A. Nguyen-Hoang, K. Nguyen, and Q.-V. Tran, "Tsgvi: a graphbased summarization system for vietnamese documents,"Journal of Ambient Intelligence and Humanized Com- puting, vol. 3, no. 4, pp. 305- 313, 2012.
7. X.-c. Ma, G.-B. Yu, and L. Ma, "Multi-document summarization using clustering algo- rithm, " in Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on, pp. 1-4, IEEE, 2009.
8. V. K. Gupta and T. J. Siddiqui, "Multi-document summarization using sentence clustering, " in Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on, pp. 1-5, IEEE, 2012
9. G. Salton, “Automatic Text Processing: the transformation, analysis, and retrieval of in- formation by computer,” AddisonWesley Publishing Company, USA, 2009.
10. Jun'ichi Fukumoto, “Multi-Document Summarization Using Document Set Type Classifi- cation,” Proceedings of NTCIR- 4, Tokyo, pp. 412-416, 2004.
11. S. Xiong and Y. Luo, "A New Approach for Multi-document Summarization Based on La- tent Semantic Analysis," Computational Intelligence and Design (ISCID), 2014 Seventh International Symposium on, Hangzhou, 2014, pp. 177-180.
12. J. Steinberger and K. Jezek, “Using latent semantic analysis in text summarization and summary evaluation,” in Proc. ISIM ’04, 2014, pp. 93–100.
13. E. Lioret and M. Palomar, 'Text summarization in progress: a literature review, " Artificial Intelligence Review, vol. 37, no. I, pp. 1-41, 2012.
14. D. Das and A. F. Martins, "A survey on automatic text summarization, "Literature Survey for the Language and Statistics II course at CMU, vol. 4, pp. 192-195, 2017.
15. S. Liu and C. G. Healey, "Abstractive Summarization of Large Document Collections Using GPT," *arXiv preprint arXiv:2310.05690*, 2023. [Online].
16. L. Ou and M. Lapata, "Context-Aware Hierarchical Merging for Long Document Summarization," *arXiv preprint arXiv:2502.00977*, 2025. [Online].
17. A. Pratapa and T. Mitamura, "Scaling Multi-Document Event Summarization: Evaluating Compression vs. Full-Text Approaches," *arXiv preprint arXiv:2502.06617*, 2025. [Online].

18. X. Ma, L. Wang, N. Yang, F. Wei, and J. Lin, "Fine-Tuning LLaMA for Multi-Stage Text Retrieval," *arXiv preprint arXiv:2310.08319*, 2023. [Online].
19. LangChain Documentation, "How to Summarize Text Through Parallelization," LangChain, 2024. [Online].
20. FAISS Documentation, "Welcome to Faiss Documentation," [Online].
21. S. A. Jaini, "Multi-Document Summarization using LLAMA2," GitHub, [Online].