

# AI-Driven Document Analysis and Reporting System for Enhanced Financial Text Processing

## Abstract

The naming of this platform is "AI-Driven Document Analysis and Reporting System", which has been designed to automatically analyze PDF Documents intelligently. The system is built using Flask for the backend and React for the frontend. This helps users interact with PDF files more easily. Users are able to summarize, interpret tables, and scope for relevant pieces of information using natural language processing. The backend processes the documents and employs AI models to provide easy-to-read summaries and insightful data. In addition, users can engage with collections that modify in real time as well as general PDFs using voice and text commands. The frontend is developed using React and TailwindCSS, returns results instantly through markdown and speech-to-text functions. Access is made simple and secure through the use of ngrok tunneling, which guarantees safe API communications. This novel system employs cutting edge structures combining AI driven NLP techniques with a user-friendly interface.

## 1 Introduction

With advances in technology, AI has transformed how we interact with papers, reports, and other documents. Documents such as legal files, financial statements, research papers, and company reports are able to be processed much faster and with a significantly lower amount of human effort, allowing productivity to skyrocket. However, the process of trying to interpret large amounts of text-based files is still extremely flawed. Extracting important insights from

documents in manual text-based formats is highly flawed as it is subjected to human error. Old-style techniques of analyzing documents are tedious as well, as it consists of reading, summarizing, and interpreting formatted data, which is completely outdated. That's where automated document processing comes in, as it can be done at scale and with a high level of precision without relying heavily on traditional methods, which result in ineffective results. These challenges along with the adoption of AI in matters like automated summarization, table understanding, and even smart querying led to the creation of an AI-Driven Document Analysis and Reporting System. The document interaction AI system, which is web based, is much more efficient in the context of productivity and incorporates natural language processing quite seamlessly. These objectives made the system intuitive and more reliant for the users enabling bypassing traditional mundane techniques and heavily relying on technology.

At the heart of the system functionality, rests the Flask implementation which acts as the core processing unit. It has API features for operations such as Document Summarization, Question Answering, and Table Analyses. The Flask Server is also connected with an AI-powered remote service which bridges communication between the system and the AI model using a ngrok URL. Besides, the backend is capable of querying several document collections, which allows users to have easier access to insights from the previously uploaded PDFs.

On the other hand, the system is built using React so that users can effortlessly upload documents and analyze them through an interactive interface. Features like dropdown-based collection selection, real-

time query responses, and web speech essence adoption make the interface easier to use and highly responsive. Furthermore, backend outputs are rendered using Markdown so that they can be structured and readable. Combining AI with a user-centric approach makes the speed and ease of document analysis more efficient, an important benefit to researchers, professionals, and other large organizations working with text data.

## 2 LITERATURE REVIEW

Document analysis and summarization has grown greatly with Large Language Models (LLMs) and deep learning techniques of Natural Language Processing (NLP). The conventional document processing systems are limited at efficiently handling unstructured data, extracting useful information in a streamlined manner, and allowing interactive queries. This “AI-Driven Document Analysis and Reporting System” proposes a solution to these issues through automating, interactivity, and scalability over PDF analysis through the application of LLMs, a flask backend, and a react based frontend. This literature review is done on previously existing research in document summarization and NLP driven table interpretation, as well as interactive document retrieval, and the new approaches to methodologies the system proposes.

### 2.1 Conventional Summarization Methods

The extractive document summarization methodologies, such as TF-IDF weighting and extractive summarization pegged on phrases and sentences using statistical sentence selection, form the earlier approaches of document summarization 7. Extractive methods are effective, but ones that would enable coherent context generation are absent, often leading to repetitive writings that are difficult to understand. To make extractive summarization more effective, attempts have been made to enhance it through the use of knowledge graph-based summarization techniques in which data relationships are used 16.

The financial and legal documents contain structured components like tables and graphs, which pose difficulties for traditional summarization models. Models like AMUSE and TIBER are built to synthesize findings from financial reports, but they give very little attention to tabular data, which is their main focus 14, 15. The requirements of a summarization system that functions with both defined and free data has led to the creation of hybrid NLP approaches.

### 2.2 LLM-Based Advances

The development of LLMs has improved summarization significantly and document splitting. FinTrAL and Instruct-Fin GPT have outperformed existing models when dealing with financial and specialized texts 1, 20. The Longformer transformer model is well-known for applying to elongated texts and has gained popularity among users dealing with large financial documents 2. RAG Blended retrieval-augmented generation has improved the accuracy of summaries by retrieving appropriate information before providing a response 3.

The “AI-Driven Document Analysis and Reporting System” combines all these technologies by applying LLMs for on-the-spot table and text analysis. This system, unlike traditional models that only focus on text extraction, boosts the efficiency of processing financial documents by summarizing reports, explaining tabular data, and allowing for multi-level QA-based insights.

### 2.3 Interactive Systems

Over a few years’ studies, various campaigns have sought to construct financial reporting tools with AI integration. FinRobot, an AI agent developed with open-source coding, performs financial reinforcement learning (FinRL) for portfolio management and predictive analytics 18. DocuBot utilizes Natural Language Processing (NLP) methodologies to create automated financial summaries from huge datasets 12. These systems, while making financial NLP more sophisticated, do not allow for interaction and real-time queries.

To mitigate this issue,<sup>19</sup> a system has been developed which integrates an interactive chatbot where users can ask queries in their natural language, obtain the main insights of documents, or give detailed table content and structured information protocols. This goes a long way in improving accessibility and user experience, making document analysis straightforward and timely.

### **3 PROBLEMS IN SUMMARIZATION AND HOW OUR SYSTEM SOLVES THEM**

#### **3.1 Interpretation of Structured Data**

Numerical and financial legal documents are always accompanied with structured tables and graphs which are beyond the reach of classic NLP models. Existing summarization tools like Tableau and other extractive techniques do not consider the underlying meaning of data stored in the tables <sup>8</sup>. The system we propose implements an LLM-based approach for table comprehension, which enables it to pull relevant structured financial data, translate it, and generate natural language explanations.

#### **3.2 Gaining Advanced User Engagement with AI Powered Queries**

Static summary generation is the main focus of coverage within most of FinTech summarizing models, which lacks interactivity. FinRobot <sup>18</sup> and DocuBot <sup>12</sup> provide the capability to build reports automatically but do not enable the user to interactively query the document. Our proposed system provides the ability to ask and interactively receive answers to questions related to the document using speech or text in real-time.

#### **3.3 Expanding Target Coverage in Combination with Hiding Summarization Contexts with Domain Specific Knowledge Gaps**

Inaccurate interpretation, at the best, is what happens when attempts are made to use general NLP techniques to comprehend these financial and legal texts <sup>9</sup>. “AI-Driven Document Analysis and Reporting System” adjusts LLMs to domain-specific developing summarizers for financial documents, thus reducing misinterpretation of documents because of poor focus.

#### **3.4 Reconfiguring Optimization to Allow Performance in Real Time and with Financial Documents Volume Constraints**

The calculations related to real-time processing of sufficiently large financial documents is a difficult problem. Improvements for extensive texts have been made using Longformer <sup>2</sup> and RAG based methods <sup>3</sup>. This project has two efforts: instantiating MySQL caching and optimizing the APIs for Flask to improve the speed of document fetching and response authoring for efficient document processing.

#### **3.5 Summarization for Decision-Making Within Seconds**

The available financial NLP models like FinGPT offer projections based only on historical data without real-time capabilities <sup>6</sup>. The suggested system automatically analyzes and summarizes financial reports to provide insights relevant to the business, legal, and research fields at the time of need.

The domain of document summarization has transformed from the simpler extractive methods to the more sophisticated LLM driven methods. The “AI Driven Document Analysis and Reporting System” leverages these techniques with multimodal summarization, real-world drilling, and data putting into context by developed insight to further improve them. This system solves the pressing issues of fi-

financial document processing in terms of accessibility, accuracy, and efficiency, which makes it useful for practitioners and researchers who regularly interact with complicated documents encapsulated in PDFs.

## 4 PROPOSED SYSTEM

The system architecture that we seek to advance has AI-based document processing at its core. Figure 1 illustrates the backend system architecture of the AI-Driven Document Analysis and Reporting System and how it is divided into various components of Flask Server such as File Processing, AI Integration and Database Management, which collectively amplify the ability to process documents.

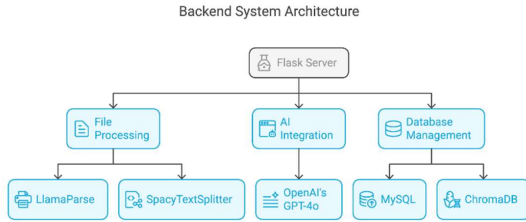


Figure 1: Backend System Architecture

### 4.1 Overview

This system intends to build AI and NLP models that will fetch and analyze information from PDF files and create document embeddings using ChromaDB. Another objective is creating a dynamic interactive user interface. The system comprises of: Frontend: A web app developed on React.js aided with Tailwind CSS for styling. Backend: A Flask service that processes files, retrieves documents, and responds to users with AI. AI NLP: Using OpenAI GPT-4o for answering questions, text summarization, and table explanations. Database: MySQL is used for meta-data and ChromaDB is used for storing vector embeddings. File Processing: LlamaParse for parsing and extracting semi-structured text from PDFs, then SpacyTextSplitter is used for splitting documents.

### 4.2 Frontend

The frontend gives users an easy-to-navigate interface for interacting with uploaded documents.

#### Features:

- **Document Upload:** Users can upload PDFs that need processing.
- **Question-Answering:** Users can pose questions and get answers in AI Chat format.
- **Table Summarization & Analysis:** Aids summaries of the documents and explains them on the tables provided.
- **Voice Queries:** They communicate through Web Speech API and respond to questions.
- **Dropdown Filter:** Users can filter and select a set of documents from a wider list of options.
- **Markdown Output:** Responses have been given in the desired format.

### 4.3 Backend (Flask)

The core of the system is the 'Flask' backend, performing document parsing, AI Question answering, and interacting with the database.

#### Primary Features:

- **PDF Handling:** In LlamaParse, we handle documents like they are a text box. This allows for text extraction and modification to Markdown.
- **Document Processing:** Documents are processed with SpacyTextSplitter and saved in vector with OpenAI Embeddings.
- **Similarity Search with ChromaDB:** A specialized database for rapid retrieval of document embeddings.
- **AI Question Answering:** Query as a document GPT-4o for use in response generation.
- **Metadata Storage:** Use MySQL to store document metadata.

## 4.4 Database & Storage

The system operates with two storage levels.

- **CSV File (data.csv):** Keeps custom file's information such as title, collection, and location of the text file.
- **MySQL Database:** Keeps AI-produced answers such as summaries and table descriptions.
- **ChromaDB:** Keeps vector embeddings for document similarity search.

## 4.5 System Workflow

- **User Uploads a PDF:** The document is uploaded by a user and is analyzed through Llama-Parse. Text that was extracted is formatted to Markdown and saved.
- **Document Chunking & Storage:** The text is segmented via SpacyTextSplitter. ChromaDB is where OpenAI Embeddings are kept after their creation.
- **User Queries the Document:** The inquiry term is performed through ChromaDB via vector search. The provided fragments are uploaded to GPT-4o for response generation.
- **AI Creates and Delivers a Response:** Once an answer is produced, it gets split into tokens and saved onto the MySQL database for later use. If the stored data is relevant, responses are served as promptly as possible.

## 4.6 Benefits of the Designed Solution

- **Enhanced Electronic Document Management:** Automation in analysis and PDF summary generation.
- **Properly Generated Insights:** Disclosure of responses through an LLM.
- **Fast and Extendable Retrieval:** It has the capability to retrieve documents using ChromaDB.

- **Attractive and User-friendly:** The application is built using React.js and the UI is designed with Tailwind CSS, making it trendy.
- **Secure and Affordable:** Embedding is done in ChromaDB with metadata kept in MySQL.
- **Everyone Can Use It:** Ideal for students in academics and research, legal professionals, and business people.

### Tech Stack:

- Flask (REST API)
- MySQL (Metadata Storage)
- ChromaDB (Vector Storage)
- LangChain & OpenAI (AI-Powered Processing)
- LlamaParse (PDF Text Extraction)
- SpacyTextSplitter (Text Chunking)

## 5 Conclusion

In conclusion, one of the ways that the AI-Driven Document Analysis and Reporting System is able to directly address the challenges concerning document analysis and processing is by incorporating AI in the form of Natural Language Processing done with Flask and React. It can summarize, extract tables, and query documents. This approach is powerful and one that will surely revolutionize the document analysis business.

Additionally, the system automates text document and data analysis, increases productivity, and facilitates advanced problem-solving. Therefore, professionals, scholars, students, and organizations that routinely analyze large volumes of textual information will be greatly aided in their productivity. Furthermore, since the system is elastic and flexible, it can be further enhanced to accommodate more document types and sophisticated functionalities.

The need for automated text document and data analysis is significant within many industries today.

Not only does this project attempt to fill in a modern-day gap, but it also lays down the foundation for AI document processing, automated knowledge extraction, or automation in document processing which makes the system exceptional. This development greatly enhances the efficiency and effectiveness of AI-controlled automation and may allow for entire sectors to fundamentally change the way data is processed for the better.

## References

- [1] Bhatia, G., Billah Nagoudi, E. M., Cavusoglu, H., Abdul-Mageed, M., The University of British Columbia, & Invertible AI. (n.d.). FINTrAL: a family of GPT-4 level multimodal financial large language models. In *arXiv:2402.10986v3 [cs.CL]* 14 Jun 2024.
- [2] Beltagy, I., Peters, M. E., Cohan, A., & Allen Institute for Artificial Intelligence. (2020). Longformer: The Long-Document Transformer. In *arXiv*. <https://arxiv.org/pdf/2004.05150.pdf>
- [3] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Haofen Wang, et al. (2024). Retrieval-Augmented Generation for Large Language Models: A survey. In *arXiv:2312.10997v5 [cs.CL]*. <https://arxiv.org/pdf/2312.10997.pdf>
- [4] Ganesan, K. (2018, March 5). ROUGE 2.0: Updated and Improved measures for evaluation of summarization tasks. *arXiv.org*. <https://arxiv.org/abs/1803.01937>
- [5] Kim, A. G., Muhn, M., Nikolaev, V. V., & University of Chicago, Booth School of Business. (2024). Financial Statement Analysis with Large Language Models. *arXiv*. <https://arxiv.org/abs/2407.17866v2>
- [6] Kong, Y., Nie, Y., Dong, X., Mulvey, J. M., Poor, H. V., Wen, Q., & Zohren, S. (2024). Large Language Models for Financial and Investment Management. *PM Research*. <https://pm-research.com/content/iijpormgmt/51/2>
- [7] Krimberg, S., Vanetik, N., & Litvak, M. (2021). Summarization of Financial Documents with TF-IDF Weighting. In *Financial Narrative Summarization (FNS 2021)*. <https://aclanthology.org/2021.fnp-1.14.pdf>
- [8] La Quatra, M., Cagliero, L., & Politecnico di Torino. (2020). End-to-End Training for Financial Report Summarization. In *FNS 2020*. <https://aclanthology.org/2020.fnp-1.20.pdf>
- [9] Litvak, M., Vanetik, N., & Puchinsky, T. (2020). SCE-SUMMARY at the FNS 2020 Shared Task. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing*. <https://aclanthology.org/2020.fnp-1.21.pdf>
- [10] Malinen, E. (2024). Interactive Document Summarizer Using LLM Technology. In *Lappeenranta-Lahti University of Technology LUT*.
- [11] Peddarapu, R. K., Anogna, S., Harshini, M. N. S. S., Sireesha, J., & Rathna, K. G. (2025). Document Summarizer: A Machine Learning Approach to PDF Summarization. *Procedia Computer Science*, 252, 1025–1036. <https://doi.org/10.1016/j.procs.2025.01.063>
- [12] Ravi, V., Amrouni, S., Stefanucci, A., Nourbakhsh, A., Reddy, P., & Veloso, M. (2020). DocuBot: Generating Financial Reports Using Natural Language Interactions. *arXiv.org*. <https://arxiv.org/abs/2010.01169>
- [13] Review of Automatic Text Summarization Techniques & Methods. (2020). *Journal of King Saud University – Computer and Information Sciences*, 34, 1029–1046. <https://doi.org/10.1016/j.jksuci.2020.05.006>
- [14] Vanetik, N., Litvak, M., & Krimberg, S. (2022). Summarization of Financial Reports

with TIBER. *Machine Learning With Applications*, 9, 100324. <https://doi.org/10.1016/j.mlwa.2022.100324>

- [15] Vanetik, N., Litvak, M., & Shamoan College of Engineering (SCE). (2021). Summarization of Financial Reports with AMUSE. <https://aclanthology.org/2021.fnp-1.5.pdf>
- [16] Vhatkar, A., Bhattacharyya, P., & Arya, K. (2020). Knowledge Graph and Deep Neural Network for Extractive Text Summarization. *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. <https://aclanthology.org/2020.fnp-1.22.pdf>
- [17] Yang, C. C., & Wang, F. L. (2003). Automatic Summarization for Financial News Delivery on Mobile Devices. In *WWW (Posters)*.
- [18] Yang, H., Zhang, B., Wang, N., Guo, C., Zhang, X., Lin, L., Wang, J., Zhou, T., Guan, M., Zhang, R., & Wang, C. D. (2024). FinRobot: An Open-Source AI Agent Platform for Financial Applications. *arXiv.org*. <https://arxiv.org/abs/2405.14767>
- [19] Yadav, N., Kanago, V., Rawat, R. S., Lambe, U. A., Musmade, S., et al. (2024). DOCSNAP.AI - AN ADVANCED DOCUMENT SUMMARIZATION TOOL. *International Journal of Creative Research Thoughts (IJCRT)*, 4, n813–n814. <https://www.ijcrt.org>
- [20] Zhang, B., Yang, H., Liu, X.-Y., Columbia University, & Swinburne University of Technology. (2023). Instruct-FINGPT: Financial Sentiment Analysis by Instruction Tuning. *arXiv:2306.12659v1 [cs.CL]*.
- [21] Zmandar, N., Singh, A., El-Haj, M., & Rayson, P. (2021). Joint Abstractive and Extractive Method for Long Financial Document Summarization. *Proceedings of the 3rd Financial Narrative Processing Workshop*. <https://aclanthology.org/2021.fnp-1.19.pdf>