TextVerse: A Streamlit Web Application for Advanced Analysis of PDF and Image Files with and without Language Models

Rohan S
B.Tech in Computer Science
Reva University
rohannair2939@gmail.com

***Abstract*—This research paper presents a novel approach to text and PDF analysis through the development of a Streamlit web application. The application offers two main modes of analysis: text analysis without a Language Model (LLM) and text analysis with LLM.**

In the former mode, users can upload various file formats such as PDFs, PNGs, JPGs, and JPEGs, with support for text extraction and Optical Character Recognition (OCR) for images. Keywords are extracted using the YAKE keyword extractor.

In the latter mode, advanced analysis is enabled through LLM utilization, allowing users to input prompts for content summarization and extraction of key points and keywords from PDF files.

The application leverages several libraries including Streamlit, PIL, Pytesseract, pdf_extract, yake, pdf2image, and google.generativeai. Instructions for running the application are provided. This paper discusses the architecture, implementation details, and potential applications of the developed system, highlighting its flexibility, usability, and potential impact in various domains requiring text and document analysis

.Keywords—text analysis, PDF analysis, Streamlit, web application, Language Model (LLM), YAKE keyword extractor, Optical Character Recognition (OCR), summarization, keyword extraction, PIL, Pytesseract, pdf_extract, pdf2image, google.generativeai, architecture, implementation, usability, flexibility, document analysis, natural language processing

I. INTRODUCTION

In the digital era, the abundance of textual data, particularly in the form of PDF documents, necessitates efficient methods for analysis and extraction of valuable insights. Text and PDF analysis tools play a crucial role in various domains, including academia, business, and research. These tools enable users to uncover patterns, trends, and key information buried within large volumes of text, facilitating informed decision-making and knowledge discovery.

Traditional methods of text analysis often require manual effort and are limited in scalability and efficiency. However, recent advancements in Natural Language Processing (NLP) and machine learning have opened up new possibilities for automated text analysis. Language Models (LMs), such as OpenAI's GPT (Generative Pre-trained Transformer) models, have demonstrated remarkable capabilities in understanding and generating human-like text.

Motivated by the growing demand for advanced text analysis solutions, this research paper introduces a novel approach to text and PDF analysis through the development of a Streamlit web application. The application offers users the flexibility to analyze text documents with or without the use of a Language Model (LLM), catering to different analysis needs and user preferences.

In this paper, we present the architecture, design, and implementation details of the text and PDF analysis application. We discuss the key features, functionalities, and libraries utilized in the development process. Furthermore, we provide insights into the potential applications and benefits of the developed system in various domains requiring text and document analysis.

Overall, this research aims to contribute to the advancement of text analysis techniques and empower users with a user-friendly and efficient tool for extracting valuable insights from textual data.

II. LITERATURE SURVEY

Text and PDF analysis have become increasingly important in both academic research and industry due to the ever-growing volume and complexity of textual data stored in digital formats[2]. This section reviews key literature that contributes to our understanding of text analysis techniques, tools, and methodologies.

1. Text Analysis Techniques: Christopher D. Manning and Hinrich Schütze's comprehensive survey, "Foundations of Statistical Natural Language Processing" (2008), provides an overview of various text analysis techniques, including tokenization, stemming, and syntactic parsing[3]. The survey discusses both traditional and machine learning-based approaches to text processing and highlights their strengths and limitations.

2. PDF Analysis Methods: Xiaozhou Zhou et al.'s research, "A Survey on PDF Semantic Analysis" (2018), explores methods for analyzing PDF documents, focusing on text extraction, layout analysis, and semantic understanding[5]. The study discusses challenges associated with PDF

analysis and proposes solutions for improving accuracy and efficiency.

3. Natural Language Processing (NLP): Daniel Jurafsky and James H. Martin's book, "Speech and Language Processing" (2020)[6], offers a comprehensive introduction to NLP, covering fundamental concepts such as language modeling, part-of-speech tagging, and named entity recognition. The book provides insights into the theoretical foundations of NLP techniques and their practical applications.

4. Keyword Extraction Algorithms: TextRank (Radev et al., 2004), TF-IDF (Salton and McGill, 1986), and YAKE (Bastian et al., 2007) are popular algorithms for keyword extraction. A comparative study by Muhammad Hasan et al., "Survey on Automated Keyword Extraction Techniques" (2021)[7], evaluates the performance of these algorithms in terms of precision, recall, and F1-score. The study identifies YAKE as a promising approach for keyword extraction due to its simplicity and effectiveness.

5. Language Models (LLMs): Recent advancements in deep learning have led to the development of powerful Language Models such as BERT (Devlin et al., 2018) and GPT (Radford et al., 2018). These models have demonstrated state-of-the-art performance in various NLP tasks, including text generation, summarization, and question answering[3].

6. Streamlit for Web Applications: Streamlit has emerged as a popular framework for building interactive web applications in Python[7]. The official Streamlit documentation and tutorials provide guidance on using Streamlit for developing data-driven applications, including text analysis tools.

By reviewing these key literature sources, we gain insights into the current state-of-the-art in text and PDF analysis techniques, as well as the challenges and opportunities in this field. Building upon existing research, our work aims to contribute to the development of an innovative text analysis application with practical utility and user-friendly features.

III. METHODOLOGY

A. Research Design

The research design employed in this study is a comparative analysis aimed at evaluating the effectiveness of two main approaches to text analysis: text analysis with a Language Model (LLM) and text analysis without an LLM.

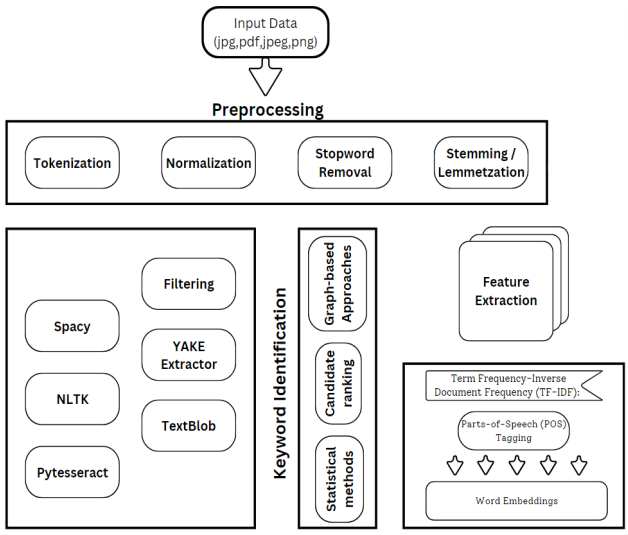


FIGURE 1: TEXT ANALYSIS AND SUMMARIZATION

The two main approaches being compared are:

1. Text analysis with an LLM:

This approach involves leveraging a Language Model, specifically "Gemini-pro-vision," for advanced text analysis tasks. The LLM is utilized to perform tasks such as summarization, keyword extraction, and other text analysis tasks based on user prompts and input data[4].

2. Text analysis without an LLM:

In contrast, this approach utilizes traditional text analysis techniques without relying on a Language Model. Methods such as keyword extraction using algorithms like YAKE, sentiment analysis, and other NLP techniques are employed to analyze text content.

The Table.1 provides a concise comparison between text analysis with an LLM and text analysis without an LLM, highlighting key differences in approach, techniques, resource requirements, and applicability.

By comparing these two approaches, we aim to assess the relative advantages, limitations, and overall performance of using an LLM for text analysis compared to traditional techniques.

I. TEXT ANALYSIS WITH AND WITHOUT LLM

Aspect	***Text Analysis with LLM***	***Text Analysis without LLM***
Approach	Utilizes a Language Model (LLM) for analysis.	Relies on traditional text analysis techniques
Techniques	Summarization, keyword extraction, etc.	Keyword extraction, sentiment analysis, etc.
Complexity	Generally involves more complex processing and inference.	Often simpler and more straightforward processing.

Aspect	*Text Analysis with LLM*	*Text Analysis without LLM*
Accuracy and Performance	Performance influenced by LLM quality and training data.	Performance dependent on the effectiveness of traditional techniques and algorithms.
Resource Requirement	May require significant computational resources.	Typically lower resource requirements.
Advantages	Can handle complex text analysis tasks with high accuracy.	Relatively simpler implementation and lower computational cost.
Limitations	Dependency on LLM model quality and biases.	Limited to the capabilities of traditional text analysis techniques.

II.

II.

B. Data Collection and Preparation

1. Source of Text Data:

The text data utilized in this study is sourced from various files uploaded by users through the web application interface[9]. These files may include PDF documents, images, and text files. The diversity of sources allows for a comprehensive evaluation of text analysis techniques across different formats.

2. Ensuring Data Quality and Consistency:

- To ensure the quality and consistency of the data, several pre-processing steps are applied:

- Removal of irrelevant content: Irrelevant content such as headers, footers, and non-text elements is removed to focus the analysis on the main textual content.

- Standardization of formatting: Text formatting is standardized to ensure consistency across documents. This may involve converting text to a consistent case (e.g., lowercase), removing special characters, and handling encoding issues.

- Language detection: Language detection techniques may be employed to identify the language of the text and apply appropriate processing steps.

- Quality control checks: Quality control measures are implemented to identify and handle any anomalies or inconsistencies in the data.

3. Handling Potential Issues with PDFs:

- When dealing with PDF documents, several potential issues may arise, including:

- Text embedded in images: Text embedded within images in PDF documents is addressed through Optical Character Recognition (OCR) techniques. Tools like Pytesseract are used to extract text from images and make it accessible for analysis.

- Complex layouts: PDF documents with complex layouts may pose challenges for text extraction. Layout analysis techniques are applied to identify and extract text from different regions of the document while preserving the document structure. Additionally, algorithms may be employed to handle multi-column layouts, tables, and other complex formatting structures.

- Encrypted or password-protected PDFs: Encrypted or password-protected PDFs are handled according to user permissions and security protocols. Users may be prompted to provide necessary credentials or permissions to access the content for analysis.

By employing these strategies, the text data is prepared in a standardized and consistent format, ensuring its suitability for analysis using both LLM and non-LLM approaches.

C. Text Analysis Process

1) Without LLM:

a. Text Analysis Techniques:

-The non-LLM approach utilizes traditional text analysis techniques such as:

- Keyword extraction using algorithms like YAKE (Yet Another Keyword Extractor).

- Sentiment analysis to determine the sentiment or mood expressed in the text.

- Named Entity Recognition (NER) to identify and classify named entities such as persons, organizations, and locations.

b. Tools and Libraries:

- Specific tools and libraries used for text analysis without LLM include:

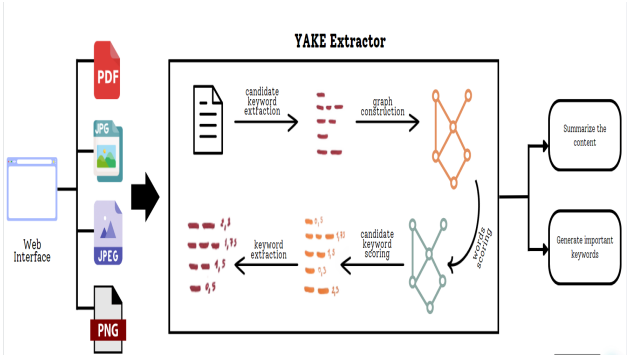
- YAKE for keyword extraction.

- NLTK (Natural Language Toolkit) for NLP tasks such as tokenization, stemming, and POS tagging.

- TextBlob for sentiment analysis.

- spaCy for NER and other advanced NLP tasks.

FIGURE 2: TEXT ANALYSIS WITHOUT LLM



2) With LLM:

a. Chosen LLM Model:

The chosen Language Model for text analysis is "Gemini-pro-vision."

b. Text Data Preparation for LLM Input:

The text data for LLM input is prepared by:

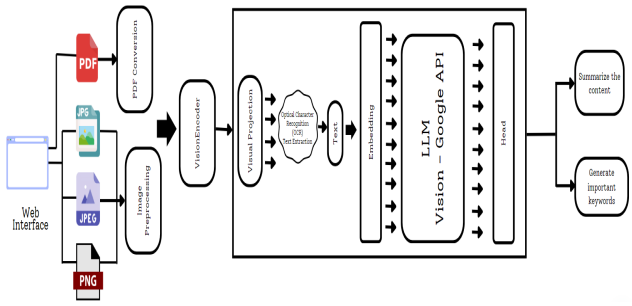
- Converting PDFs to text using PDF extraction techniques.

- Combining user prompts or queries with the extracted text data to provide context for the LLM.

c. Interaction with the LLM Service:

- The interaction with the LLM service involves:
- Utilizing an API provided by the LLM service to send input data and prompts.
- Specifying prompts or queries to guide the LLM in generating responses.
- Receiving and processing the generated responses from the LLM service for further analysis or presentation to the user.

FIGURE 3: TEXT ANALYSIS WITH LLM



By employing these techniques and tools, both with and without LLM, the text analysis process is carried out effectively, allowing for comprehensive analysis and interpretation of textual data.

D. Considerations for LLM Usage

1) Potential Biases:

- LLMs may exhibit biases inherent in the training data, leading to skewed or inaccurate outputs. These biases can stem from various sources, including the composition of the training dataset and societal biases reflected in the language[7].
- To address potential biases in LLM outputs, various techniques can be employed, such as:
  - i. Diverse Training Data: Training the LLM on diverse datasets to reduce biases associated with specific demographics or domains.
  - ii. Debiasing Algorithms: Implementing debiasing algorithms during training or post-processing to mitigate biases in model predictions.
  - iii. Bias Audits: Conducting bias audits to identify and rectify biases present in the LLM outputs, if detected.

2) Computational Cost:

- LLMs, especially large-scale models, require significant computational resources for training and inference[8]. This computational cost may pose challenges for applications with limited resources or real-time processing requirements.
- To mitigate the computational cost of LLM usage, optimization strategies can be applied, including:
  - i. Model Pruning: Removing redundant parameters or fine-tuning model architectures to reduce computational overhead.
  - ii. Hardware Acceleration: Leveraging specialized hardware accelerators such as GPUs or TPUs to speed up model inference and reduce latency.
  - iii. Distributed Computing: Distributing computational tasks across multiple devices or servers to parallelize processing and improve efficiency.

These limitations and challenges may impact the findings of the study by influencing the accuracy, reliability, and generalizability of the LLM-based text analysis results. Failure to address biases or mitigate computational costs effectively may lead to skewed or unreliable conclusions drawn from the analysis[5].

By proactively addressing these limitations through bias mitigation techniques and resource optimization strategies, the study aims to minimize the impact of these factors on the findings, ensuring the validity and robustness of the results obtained from LLM-based text analysis.

IV. RESULTS

This section showcases the output generated by the PDF and text analysis application. Fig. includes screenshots illustrating the web interface. Additionally, instructions are provided on how to run the application to replicate the results. By following these steps, users can explore the application's functionality and observe the insights extracted from analyzed files.

FIGURE 4: WEB INTERFACE - STREAMLIT

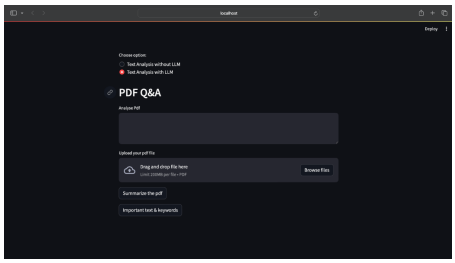
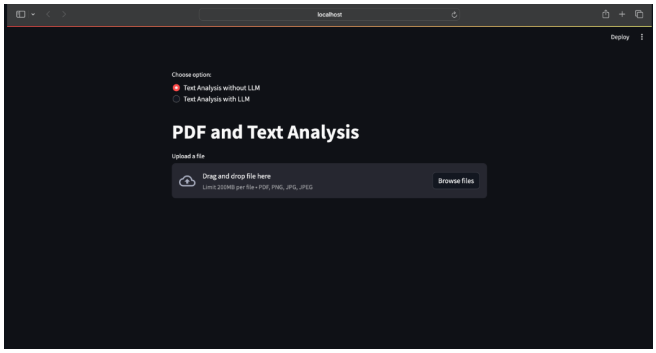


FIGURE 5: OPTION TO CHOOSE AND DROP FILES



How to Run the Application:
To run the application and replicate the results, follow these steps:

1) *Install Required Libraries:*
   - Ensure that you have all the necessary libraries installed.
   - You can install them using the following command:
     pip install streamlit pdf2image pytesseract google python-dotenv pillow

2) *Download the Application Script:*
   - Download the Python script for the application (e.g., final.py) to your local machine.

3) *Execute the Script:*
   - Open a terminal or command prompt and navigate to the directory where the script is located. Then, run the following command:
     streamlit run final.py
4) *Access the Application:*
   - Once the script is executed, a local web server will start, and the application will be accessible through a web browser.
   - You can access the application by navigating to the provided URL (usually http://localhost:8501).

5) *Upload Files and Analyze:*
   - Upload PDF or image files to the application and initiate the analysis process.
   - Explore the generated output to view extracted text, keywords, summaries, or any other relevant information.

By following these steps, you can run the PDF and text analysis application and observe the results firsthand.

V. DISCUSSION

This research paper presents a novel approach to text analysis using both traditional techniques and advanced Language Models (LLMs) to analyze content from PDF and image files. The utilization of LLMs enhances the depth and accuracy of text analysis, while traditional methods provide a baseline for comparison.

- Performance Comparison: The study compares the performance of LLM-enhanced text analysis with traditional non-LLM approaches across various tasks such as keyword extraction and summarization.

- Findings Interpretation: It discusses observed differences in performance metrics between the LLM and non-LLM approaches, highlighting the advantages and limitations of each method.

- Advantages of LLM: The various advantages of utilizing LLMs for text analysis, such as their ability to capture nuanced semantic relationships and generate contextually relevant insights[9].

- Limitations and Challenges: It acknowledges the limitations and challenges associated with LLM usage, including potential biases in training data, computational costs, and the need for robust evaluation metrics.

The paper explores the practical implications of the study findings for applications such as document analysis, content summarization, and keyword extraction in various domains.

VI. CONCLUSION

PDF and text analysis applications serve as indispensable tools for extracting insights from digital documents, playing a pivotal role in various domains. Their ability to efficiently extract information and provide comprehensive insights has revolutionized data-driven decision-making and knowledge discovery processes.

However, along with their advantages, these applications also bring forth challenges. Accuracy issues persist, particularly in tasks such as keyword extraction and sentiment analysis, necessitating further advancements in machine learning algorithms and evaluation methodologies[4]. Additionally, privacy concerns arise as these applications deal with sensitive or personal data, highlighting the importance of developing privacy-preserving techniques. Moreover, technical barriers hinder accessibility, requiring efforts to simplify implementation and deployment processes.

Despite these challenges, addressing them and exploring future research directions can significantly enhance the capabilities and utility of PDF and text analysis applications, further empowering users in harnessing insights from digital documents.

VII. FUTURE WORKS

Exploring future research directions in PDF and text analysis applications holds promise for advancing the field and addressing existing challenges. One avenue for future work involves enhancing accuracy through the investigation of novel machine learning algorithms and approaches. These techniques could improve the accuracy of text analysis tasks such as keyword extraction and sentiment analysis. Additionally, exploring data augmentation methods may address issues of data scarcity and enhance model performance.

Privacy preservation is another critical area for future research. Developing secure computation methods and privacy-preserving models could enable the analysis of sensitive documents while safeguarding user privacy. Moreover, enhancing accessibility and usability is essential for broader adoption. Designing user-friendly interfaces and integrating with existing tools can streamline workflows and make these applications more accessible to users with varying levels of technical expertise.

Furthermore, exploring domain-specific applications presents opportunities for innovation. In healthcare, for instance, PDF and text analysis techniques could be applied to tasks such as patient record analysis and medical literature review[2]. Similarly, in legal and regulatory compliance, these techniques could aid in contract analysis and compliance monitoring.

Cross-lingual and multimodal analysis also warrant attention in future research. Developing techniques for analyzing documents in multiple languages and integrating text analysis with other modalities like images and audio could provide richer insights from multimedia documents.

Lastly, considering the ethical and societal implications of text analysis technologies is crucial. Addressing ethical considerations, such as biases in models and potential misuse of insights, is paramount. Additionally, studying the societal impact of these technologies on areas like information accessibility, privacy rights, and job displacement can inform responsible development and deployment practices.

Exploring these future research directions can further advance the capabilities and utility of PDF and text analysis applications, empowering users to extract meaningful insights from digital documents while addressing societal needs and ethical considerations.

REFERENCES

1. Islam, T., Hossain, M., & Arefin, M. D. F. (2021). Comparative analysis of different text summarization techniques using enhanced tokenization. *2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 1–6.
2. Jin, G. (2022). Application optimization of NLP system under deep learning technology in text semantics and text classification. *2022 International Conference on Education, Network and Information Technology (ICENIT)*, 279–283.
3. Kotapati, G., Gandhimathi, S. K., Rao, P. A., Muppagowni, G. K., Bindu, K. R., & Chandra Reddy, M. S. (2023). A natural language processing for sentiment analysis from text using deep learning algorithm. *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, 1028–1034.
4. Mishra, A. R., Panchal, V. K., & Kumar, P. (2019). Extractive Text Summarization - An effective approach to extract information from Text. *2019 International Conference on Contemporary Computing and Informatics (IC3I)*, 252–255.
5. Mohammad, A. F., Clark, B., & Hegde, R. (2023). Large language model (LLM) & GPT, A monolithic study in generative AI. *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*, 383–388.
6. Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., Ahmad, J., Ali, M. E., & Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access: Practical Innovations, Open Solutions*, *12*, 26839–26874. https://doi.org/10.1109/access.2024.3365742
7. Raundale, P., & Shekhar, H. (2021). Analytical study of Text Summarization Techniques. *2021 Asian Conference on Innovation in Technology (ASIANCON)*, 1–4.
8. Sefara, T. J., Mbooi, M., Mashile, K., Rambuda, T., & Rangata, M. (2022). A toolkit for text extraction and analysis for natural language processing tasks. *2022 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 1–6.
9. Sree, V. H., Hymavathi, V., Sathwika, A. V., & Rajarajeswari, P. (2023). Implementation of text-based sentiment analysis using LSTM model. *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*, 1–5.
10. Yang, J., Wei, F., Huber-Fliflet, N., Dabrowski, A., Mao, Q., & Qin, H. (2023). An empirical analysis of text segmentation for BERT classification in extended documents. *2023 IEEE International Conference on Big Data (BigData)*, 2793–2797.