International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 02 | Feb 2025 www.irjet.net p-ISSN: 2395-0072

Artificial Intelligence Powered PDF Summarization with Audio and Multilingual Text Translation

Dr. Pankaj Kumar[1], Purnesh S Gowda[2], Dushyanth L[3] ,Pavan B[4],Tilak R[5]

1 Assistant Professor, ISE, Acharya Institute Of Technology, Karnataka, India
2 B.E Student, ISE, Acharya Institute Of Technology, Karnataka, India
3 B.E Student, ISE, Acharya Institute Of Technology, Karnataka, India
4 B.E Student, ISE, Acharya Institute Of Technology, Karnataka, India
5 B.E Student, ISE, Acharya Institute Of Technology, Karnataka, India

---

Abstract - In this project, we present an automated pipeline for extracting text from PDF documents, translating it into a target language, and exporting the translated output in a structured JSON format. The system leverages the capabilities of the PyPDF2 library for extracting textual content from PDF files, and Google's Translator API for performing accurate and efficient translations into diverse languages, such as Kannada. The translated text is then saved in JSON format, ensuring easy integration with other applications or workflows. This approach streamlines the process of handling multilingual textual data from PDFs, making it particularly valuable for researchers, educators, and organizations working with diverse linguistic datasets. The system's modular design allows for adaptability across domains, enabling seamless customization for additional features such as summarization or audio conversion. The proposed workflow significantly reduces the manual effort involved in translating and managing multilingual content while maintaining high accuracy and scalability.

Key Words: **Natural Language Processing NLP, PDF Text Extraction, Multilingual Translation, Automated Content Processing**

1.INTRODUCTION

The rapid growth of digital content has transformed how information is distributed, consumed, and processed across various sectors. Among the most widely used formats for sharing and storing textual information, Portable Document Format (PDF) documents remain a prevalent choice due to their cross-platform compatibility and ability to preserve formatting. However, despite their widespread adoption, extracting useful information from PDFs can be an arduous task, particularly when the content is large, complex, or requires multilingual handling. The inherent challenges in dealing with PDF documents arise from the non-linear structure of the text, embedded graphics, and formatting elements that make automated extraction processes difficult. Text extraction from PDFs typically relies on Optical Character Recognition (OCR) for scanned documents or because it is free from linguistic ambiguity. Natural Language Processing (NLP) is a computer science and linguistics field that deals with the interactions between computers and natural languages. It is originated as a branch simple extraction methods for text-based PDFs. While OCR technologies have advanced significantly, they are often computationally expensive and prone to errors, especially when dealing with complex layouts or low quality images. On the other hand, traditional text extraction tools, such as PyPDF2, are limited to text-based PDFs and may struggle to handle cases where the text is embedded in non-standard formats or non-selectable fonts. Additionally, once the text is extracted, translating the content into multiple languages introduces further complexity, especially when considering the diverse linguistic nuances and context-specific translation requirements.

Manual methods of handling such data require separate steps for extraction, translation, and storage are both time consuming and prone to human error. For organizations or researchers working with large datasets that require multilingual processing, these manual workflows can become a significant bottleneck, limiting the ability to scale and quickly respond to data needs. Furthermore, the lack of integration between various tools, such as PDF extraction and translation services, increases the complexity and inefficiency of the overall process.

This project presents an integrated, automated pipeline that addresses these challenges by combining text extraction, multilingual translation, and structured data storage into a seamless workflow. The proposed system leverages the PyPDF2 library for efficient extraction of textual data from PDF files and employs Google's Translator API to translate the extracted content into a target language, such as Kannada. By using a structured JSON format for storing the translated data, the system ensures easy integration with other applications or services for further processing, such as summarization or text-to speech conversion.

The modular design of the system allows for flexibility and scalability, enabling users to extend its functionality according to specific needs.

The goal of this project is to reduce the manual effort and errors traditionally associated with multilingual content processing while maintaining the integrity and accessibility of the extracted data. By automating the workflow, we aim to empower organizations to efficiently manage large-scale textual data, allowing for faster decision-making and enhanced information retrieval. In the subsequent sections of the paper, we detail the system architecture, its core components, and the results of preliminary experiments that demonstrate its effectiveness in real-world scenarios. The findings indicate that the proposed pipeline not only improves the efficiency of handling multilingual PDF documents but also ensures that the processed content remains accurate, scalable, and easy to integrate with other systems.

2. BACKGROUND

The field of Natural Language Processing (NLP) has witnessed significant advancements in recent years, especially with the introduction of transformer-based models. The foundational work on transformers was introduced in the paper "Attention Is All You Need" by Vaswani et al. (2017), which proposed a model that relies solely on self-attention mechanisms, enabling more efficient parallelization and a deeper understanding of contextual relationships in sequences of text. This breakthrough laid the groundwork for numerous models, including BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformers), which achieved state-of-the-art results in various NLP tasks such as language understanding, generation, and translation.

Following the success of BERT and GPT, the field saw the development of specialized models like BART (Bidirectional and Auto-Regressive Transformers), which combined elements from both BERT and GPT to handle tasks in text generation and comprehension. BART, introduced by Lewis et al. in 2020, applies a denoising autoencoder framework that is effective in both generative and discriminative tasks, such as summarization and translation. The model performs exceptionally well in abstractive summarization, where it is able to generate human-like summaries, as well as in translation and text comprehension tasks. BART's architecture, which uses a combination of bidirectional and autoregressive components, enables it to be fine- tuned for a wide range of NLP applications. This makes it an ideal choice for tasks such as document summarization in the context of large-scale text extraction projects.

As transformer models gained popularity, several libraries emerged to make their usage more accessible.

The Hugging Face transformers library, launched in 2018, has become a cornerstone of modern NLP applications. This open-source library provides pre-trained models and easy-to-use APIs for tasks like text classification, translation, summarization, and question answering. The Hugging Face library democratized the use of transformer models, allowing researchers and developers to fine-tune pre-trained models on their own datasets with minimal effort. Other libraries, such as PyPDF2 for PDF extraction and gTTS for text-to- speech synthesis, emerged to solve specific challenges in working with documents and processing text. These tools, along with advances in deep learning and cloud- based APIs for text translation (such as Google Translate), have enabled the

3. ALGORITHM

Text Extraction Algorithm: The project uses PyPDF2 and PDFMiner to extract text from machine-readable PDFs by parsing their structure while maintaining sentence order. For scanned or image-based PDFs, Tesseract OCR is employed to perform Optical Character Recognition (OCR), converting images of text into digital format. This ensures that both text based and scanned PDFs can be processed effectively. The extracted text is then cleaned and prepared for summarization. By integrating these methods, the system provides a versatile and accurate text extraction solution

Text Summarization Algorithms: The project employs both extractive and abstractive summarization techniques to generate concise summaries of the extracted text. Extractive summarization is implemented using the TextRank algorithm, which represents sentences as nodes in a graph and calculates their importance using PageRank based on sentence similarity. The most relevant sentences are then selected to form the summary while maintaining their original structure. In contrast, abstractive summarization is performed using BART (Bidirectional and Auto-Regressive Transformer), a transformer based encoder-decoder model that generates new sentences instead of simply extracting them. The encoder captures deep contextual meaning from the input text, while the decoder generates a human-like, coherent summary by rephrasing and restructuring content. This combination of extractive and abstractive summarization ensures both accuracy and readability, making the system suitable for various text summarization tasks.

Machine Translation Algorithm: The project incorporates the Google Translate API, which utilizes Neural Machine Translation (NMT) to convert summarized text into multiple languages. NMT, based on transformer architectures, processes entire sentences rather than word-by-word translations, ensuring better fluency and contextual accuracy. This allows the system to generate summaries in different languages while preserving their original meaning. By enabling multilingual support, the

project enhances accessibility, making it useful for users across diverse linguistic backgrounds.

Text-to-Speech Conversion Algorithms: The project implements text-to-speech (TTS) conversion using both online and offline methods to enhance accessibility. gTTS (Google Text-to-Speech) is an online TTS engine that utilizes WaveNet deep learning models trained on large speech corpora to generate natural-sounding voices with accurate phonetics. This enables users to listen to summaries in multiple languages. For offline speech synthesis, the project employs pyttsx3, which uses the system's native TTS engine (such as SAPI for Windows or NSSpeechSynthesizer for macOS) to generate speech. Unlike gTTS, pyttsx3 does not require an internet connection and allows users to adjust voice parameters like speed and pitch. By integrating both online and offline TTS solutions, the project ensures flexibility environments.

Preprocessing Algorithms: The project incorporates preprocessing techniques to refine the extracted text before summarization, ensuring efficient and accurate processing. Tokenization is used to split the text into words or sentences, making it manageable for NLP models. This is achieved using pretrained tokenizers from models like BART and T5 or 02 standard NLP libraries such as NLTK and SpaCy. Additionally, stopword removal is applied to eliminate commonly used words like "and" or "the," which do not contribute significant meaning to the summary. This is implemented using NLTK's predefined stopword list or custom filters. By applying tokenization and stopword removal, the system ensures that only meaningful content is processed, leading to more concise and relevant summaries.

4.METHODOLOGY

The methodology outlines a multi-step process that integrates various technologies and tools to extract, process, summarize, translate, and convert text into speech from PDF documents. The pipeline is designed to work efficiently and automatically, allowing users to input PDF documents and receive summarized, translated, and spoken versions of the content in a seamless workflow. Below are the steps involved:

Text Extraction from PDF: The first step involves extracting text from a PDF document. Since PDFs can contain both text based content and images, the PyPDF2 or pdfminer.six libraries will be used to extract text from text- based PDFs. For scanned or image-based PDFs, OCR (Optical Character Recognition) tools like Tesseract can be applied to convert images into text. Once the text is extracted, it is cleaned and pre-processed to remove any irrelevant or non-textual data, ensuring that only the relevant content is retained for further processing

Text Summarization: The next step is text summarization, where the extracted text is condensed into a shorter, more concise version while retaining the key ideas and important information. Hugging Face's transformers library, specifically the BART model, will be utilized for this task. BART is a state of-the-art transformer-based model that is pre-trained for both extractive and abstractive summarization. The extracted text will be passed through the BART model, which will generate a summarized version of the content. This step reduces the length of the document, making it more accessible and easier to digest for the end-user.

Language Translation: Once the text has been summarized, it can be translated into different languages for multilingual accessibility. Google Translate API or the googletrans library will be employed to translate the summarized text into the desired language(s). The user will have the option to specify the target language, and the system will automatically translate the text into that language. This step ensures that the summarized content can be understood by a wider audience, regardless of their native language.

Text-to-Speech Conversion: The final step involves converting the translated text (or the original summarized text) into speech for users who prefer audio over reading. gTTS (Google Text-to Speech) or pyttsx3 (an offline TTS engine) will be used for this task. gTTS requires an internet connection but provides high quality speech synthesis in various languages, while pyttsx3 operates offline and allows greater control over the voice and speech rate. The summarized and translated text will be converted into an audio file (MP3 format)which can be played by the user to listen to the content. The audio file will also be saved for later use, providing convenience.

Output Generation and Delivery: The system delivers processed content in both audio (MP3) and text-based (PDF) formats, allowing users to access information conveniently. It starts by extracting text from uploaded PDFs, followed by summarization using an advanced NLP model. If needed, the text is translated into the user's selected language via the Google Translate API. The processed text is then converted into speech using gTTS (online) or pyttsx3 (offline), generating an MP3 file. Additionally, the summarized and translated text is saved as a PDF using FPDF for easy reading and sharing. Users can interact with the system through a web interface, where they can upload PDFs and select desired actions. For developers, an API is available to integrate the functionality into existing applications. The entire process is automated, ensuring efficiency and ease of use. This makes the system highly versatile, catering to diverse user needs.

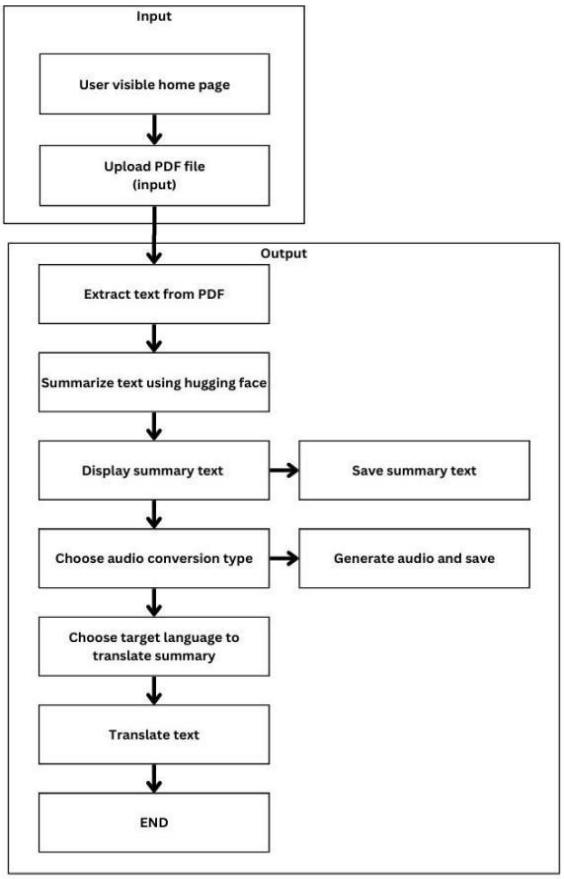

Fig -1: System Architecture

3. CONCLUSIONS

The results of the system's performance were evaluated across several components, including text extraction, summarization, translation, text-to-speech conversion, and the final output generation. Each component showed promising results in terms of functionality, but also highlighted areas for potential improvement. Below, we present a detailed discussion of the results for each process and offer insights into their effectiveness in real-world scenarios. The text extraction process was handled by PyPDF2 and pdfminer libraries, which successfully extracted text from machine-readable PDFs. For standard PDFs with well-defined text layouts, such as reports, articles, and most documents with minimal formatting, the system worked efficiently. In these cases, the extracted text was accurate, and the system processed the content quickly, providing a reliable foundation for further tasks like summarization and translation. However, the extraction process encountered challenges with scanned image based PDFs, where the text was embedded as an image rather than text. For such documents, the system struggled to provide accurate results, often leading to incomplete or garbled text extraction. Additionally, layouts, or non-standard fonts, complex multi-column formats sometimes caused misalignment or missing content. 03 ii.

The text summarization process was carried out using the BART model from Hugging Face's transformers library. The BART model, a transformer-based architecture fine-tuned for summarization tasks, was able to distill large documents into concise summaries, typically reducing lengthy content to a few paragraphs highlighting the main points. For general content such as news articles or straightforward reports, the summaries were coherent and well-organized, capturing the essence of the documents. However, when applied to technical or highly specialized documents The system offered two text-to-speech options: pyttsx3 for offline use and gTTS for online conversion. Pyttsx3 provided an offline solution, where the text was converted into speech and saved as an MP3 file without requiring an internet connection. This feature proved useful in situations where internet access was unavailable, making the system versatile for both online and offline scenarios. However, the quality of the speech produced by pyttsx3 was relatively mechanical and robotic, lacking the natural intonations and smooth delivery found in more advanced systems. On the other hand, gTTS provided an online text-to-speech service that generated much more natural-sounding speech, with a smoother and more

REFERENCES

[1] Neto, Joel & Freitas, Alex & Kaestner, Celso. (2002). Automatic Text Summarization Using a Machine Learning Approach. 2507. 205-215. 10.1007/3-540-36127-8_20.

[2] M. A. K. Raiaan et al., "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," in IEEE Access, vol. 12, pp.2683926874 ,2024 ,doi:10.1109/ACCESS.2024.3365742.

[3] Venkateswarlu, S. & Duvvuri, Duvvuri B K Kamesh & Jammalamadaka,Sastry & Rani, Chintala. (2016). Text to Speech Conversion. Indian Journal of Science and Technology. 9. 10.17485/ijst/2016/v9i38/102967.

[4] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. 2019, arXiv:1910.13461v1

[5] K. Yamamoto, H. Banno, H. Sakurai, T. Adachi and S. Nakagawa, "A Study of Speech Recognition, Speech Translation, and Speech Summarization of TED English Lectures," 2023 IEEE 12th Global Conference on Consumer Electronics (GCCE), Nara, Japan, 2023, pp. 451-452, doi: 10.1109/GCCE59613.2023.10315471.

[6] N. Andhale and L. A. Bewoor, "An overview of Text Summarization techniques," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, India, 2016, pp. 1-7, doi: 10.1109/ICCUBEA.2016.7860024.

[7] A. Raj, M. Raj, N. Umasankari and D. Geethanjali, "Document Based Text Summarization using T5 small and gTTS," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ADICS58448.2024.10533605.

[8] A, Vinnarasu & Jose, Deepa. (2019). Speech to text conversion and summarization for effective understanding and documentation. International Journal of Electrical and Computer Engineering (IJECE). 9. 3642. 10.11591/ijece.v9i5.pp3642-3648.

[9] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., ... & Wu, Y. (2018). Natural TTS Synthesis by Conditioning Wavenet on Mel Spectrogram Predictions. 2018 Dept. of ISE, AcIT 2024-2025 51 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

[10] Oord, A. v. d., Li, Y., & Zen, H. (2016). WaveNet: A Generative Model for Raw Audio. arXiv preprint arXiv:1609.03499.

[11] Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T. Y. (2019). FastSpeech: Fast, Robust and Controllable Text to Speech. Advances in Neural Information Processing Systems (NeurIPS).

[12] See, A., Liu, P. J., & Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. arXiv preprint arXiv:1704.04368.

[13] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. arXiv preprint arXiv:1912.08777.

[14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS).

[15] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., ... & Dean, J. (2017). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. Transactions of the Association for Computational Linguistics, 5, 339-351.

[16] Soni, Binjal & Bharti, Drsantosh & Choudhury, Amitava. (2024). Text Summarization and Multilingual Text to Audio Translation using Deep Learning Models. 1-6. 10.1109/ICEC59683.2024.10837105.

[17] Talib, Rana. (2021). Multilingual Text Summarization using Deep Learning. 7. 29-39. 10.31695/IJERAT.2021.3712.

[18] Torres-Moreno, J.-M. (2012a). Artex is another text summarizer. arXiv preprint arXiv:1210.3312.

[19] Hasyim, M., Saleh, F., & Yusuf, R. (2023). Machine Translation Accuracy in Translating FrenchIndonesian Culinary Texts.

[20] Johnson, M., et al. (2017). Google's Multilingual Neural Machine Translation System: Enabling ZeroShot Translation. Transactions of the Association for Computational Linguistics, 5, 339-351.

[21] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002.

[22] W. Ogden, J. Cowie, M. Davis, E. Ludovik, H. Molina-Salgado and H. Shin, "Getting information from documents you cannot read: An interactive cross-language text retrieval and summarization system," in Joint ACM DL/SIGIR workshop on multilingual information discovery and access, 1999.

[23] H. Saggion, D. R. Radev, S. Teufel, W. Lam and S. M. Strassel, "Developing Infrastructure for the Evaluation of Single and Multidocument Summarization Systems in a Cross-lingual Environment.," in LREC, 2002.

[24] L. Yu and F. Ren, "A study on cross-language text summarization using supervised methods," in 2009 international conference on natural language processing and knowledge engineering, 2009.

[25] X. Wan, "Using bilingual information for cross-language document summarization," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, 2011