BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*
(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)
"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India) Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

2023-2024

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Project Work Phase -1 Synopsis

"PPTFUSSION- a web application to convert PDF documents into PowerPoint presentations."

Submitted By

B SAI KIRAN REDDY	**USN: 3BR20CS022**
B SAI TEJESWINI	**USN: 3BR20CS030**
C PALLAVI	**USN: 3BR20CS035**
G ANJANASREE	**USN: 3BR20CS047**

Under the Guidance of

Mr. Phaniram Prasad
Assistant Professor
Dept of CSE
BITM, Ballari.

Visvesvaraya Technological University

Belagavi, Karnataka 2023-24

ABSTRACT

This application, named "pptfussion" is a Flask-based web application that primarily serves to convert PDF documents into PowerPoint (PPTX) presentations. The application allows users to upload a PDF file, and it then extracts headings and content from the PDF. These extracted sections are then used to generate a PowerPoint presentation with images, where each slide represents a section from the PDF, with headings as slide titles and summarized content from the PDF as slide content. This innovative application addresses the growing need for efficient content repurposing, making it invaluable for professionals, educators, and students. In an era where information is abundant, this application offers an efficient solution for content repurposing, helping users transform static documents into engaging presentations and distil lengthy texts into digestible summaries. Whether for business presentations, academic reports, or research papers, the pptfussion Application empowers users to save time and enhance the impact of their content. Thus, collaborating with unparalleled ease and efficiency for the user.

VISION

The vision of "pptfussion" is to simplify the process of converting lengthy PDF documents into concise and visually appealing PowerPoint presentations and providing a user-friendly interface for this task.

MISSION

Our mission is to revolutionize the way individuals handle information by providing a cutting-edge application that seamlessly converts PDF documents into dynamic PowerPoint presentations and transforms lengthy textual content into concise, informative summaries. Thus, to provide a user-friendly, efficient, and versatile application that enhances the way people communicate and present information. We are committed to enhancing productivity, facilitating knowledge sharing, and empowering users to effectively communicate and repurpose content. Our goal is to simplify the content adaptation process and make information more accessible and engaging for a global audience.

OBJECTIVES

1. To simplify the process of PDF-to-PPTX conversion.
2. To automatically extract headings and content from PDF documents.
3. To generate PowerPoint presentations with slide titles, summarized content and effective images.
4. To provide users with the option to download the resulting PowerPoint presentation.

PROBLEM STATEMENT

Converting lengthy textual PDF documents into attractive PowerPoint presentations and providing a user-friendly interface for this task.

SCOPE OF THE PROJECT

The scope of the "pptfussion" project is to provide a web-based platform for users to upload PDF files and receive corresponding PowerPoint presentations. The application's focus is on converting textual PDFs with clear headings into concise presentations. It does not handle complex formatting or multimedia elements. The project's scope includes streamlining the process, ensuring usability, and delivering a reliable conversion service.

LITERATURE SURVEY

The literature survey for "pptfussion" may include research and resources related to PDF-to-PPTX conversion, natural language processing techniques for summarization, and web application development with Flask. It involves studying relevant academic papers, articles, and software tools to understand the state of the art in the field and incorporate best practices into the application's development.

In paper[1], the author presented an approach Extracting text from a PDF file is a task that sounds easier than its real-life execution. PDF files namely only know the position of the characters on the page, not knowing that the characters form words together. Another challenge is to separate it into different sections and paragraphs. This research paper focuses on text extraction from a PDF-formatted CTI report, intending to extract the text separated into the sections present in the CTI report. This paper contained information analyzing multiple candidate tools, which text extraction tool is preferred for text and section extraction from a PDF file using Python. This paper carried information on various candidate tools like PDFPlumber (a tool is capable of text extraction from a PDF file), PymuPDF (tool that allows for the extraction of text from a PDF file preserving the layout of the document), PyPDF2 (pure-python tool and capable of extracting metadata).

In paper[2], the author presented an approach this paper briefed about how the Web is the most frequently and rapidly used networking aid which satisfies the requirements of all types of users, and which provides a solution for any type of problem. For designing and developing such well-defined and well structured, we have to choose a proper technology. Therefore, a dynamic web application or portal can be developed by using flask and python. Good development of a web page or an application can easily attract users which leads to success of the project. The technological needs of a web development project can be achieved by using “python” and “flask”. Flask is a web framework. It is a lightweight web application framework written in Python and baseband on the WSGI toolkit and Jinja2 template engine. This means flask provides you with tools, libraries and technologies that allow you to build a web application. Thus, introducing us to the concepts of flask its robust and open-source nature

In paper[3], the author presented an approach this paper briefed about how the exponential increase of textual information on the internet has led to a considerable expansion of digital content. However, this abundance of information makes it challenging to extract valuable insights due to the sheer volume of content. Text summarization has become an essential tool to address this issue by providing a condensed version of the selected content. This research paper introduces an Auto Text Summarizer Application is introduced which is developed in Python. The application can accept a web page URL or textual input as its source, which is then processed to generate a summary using the Extractive Text Summarization technique. The application utilizes four distinct Python libraries including Natural Language Toolkit (NLTK), Spacy, Gensim, and Sumy, and Flask framework is employed to present the summarized content on the front-end. The back end of the application involves the use of the Beautiful Soup library to scrape web page content or read the provided text data. The results obtained by each of these libraries are

compared based on the reading time required for the summarized content, while also computing Rouge Score, F1 Score and Precision. The development of the Text Summarizer Application is a valuable addition to the Natural Language Processing domain, as it provides a means for summarizing large volumes of textual data in an efficient and effective manner. Furthermore, the use of Python libraries and frameworks makes this application scalable and easy to use, while also providing accurate and reliable results.

In paper[4], the author presented an approach the paper contained information about which is a pure Python library that fights on two fronts:
1) digital document indexing and similarity search; and 2) fast, memory-efficient, scalable algorithms for Singular Value Decomposition and Latent Allocation. The connection between the two is unsupervised, semantic analysis of plain text in digital collections. It was created for large digital libraries, but its underlying algorithms for large-scale, distributed, online SVD and LDA are like the Swiss Army knife of data analysis—also useful on their own, outside of the domain of Natural Language Processing.

In paper[5], the author presented an approach this paper explored an exceptionally difficult assignment of consequently creating presentation slides for scholarly papers. The created presentation slides can be utilized as drafts to offer the moderators some assistance with preparing their formal slides quickerly. A novel framework called PPSGen is proposed to address this undertaking. It first utilizes the relapse strategy to take in the significance scores of the sentences in a scholarly paper, and afterward misuses the whole number straight programming (ILP) system to produce very much organized slides by selecting and adjusting key expressions and sentences. Assessment results on a test set of 200 sets of papers and slides gathered on the web show that our proposed PPSGen framework can produce slides with better quality.

In paper[6], the author presented an approach Automatic Text Summarization is still today a very active research topic with several methods available and hundreds of applications, summarization from scientific papers is one of them. In general, text Summarization can be extractive or abstractive. This work will focus only on extractive methods; therefore the summaries will be composed of some sentences taken from the original text and re-arranged together. The output from the automatic summarization of a scientific paper can be used to create a presentation for a scientific work i.e., slides or can be read to understand the main contents and gain more information than simply reading the Abstract and conclusions. It addresses the following tasks:

- text summarization, based on a Seq2seq model (Recurrent Neural Networks used for text paraphrasing).
- figure extraction, which entails identifying the most relevant figures in the paper by using a shared embedding space for images and text. Images are embedded using a ResNet-152 and text is encoded using RoBERTa (a Transformer model, read below for details), then the embeddings are projected to a shared embedding space using a two-layer multilayer perceptron (MLP).
- slide generation, which relies on a module that is able to put paraphrased objects (text and/or images) in a visually appealing manner using a two-layer MLP to predict the position and size of the object.

In paper[7], the author presented an approach the automatic generation of presentation slides from technical articles is one of the most desired but under-researched areas in the field of computing. Automated generation of slide con tents from technical articles is much difficult than a typical text summarization process, since it requires the identification of all the crucial contents from the article and their arrangement in a systematic manner, thus making it a non-trivial task. The process is considered to be one of the core applications of text mining. Automatic slide generators can be broadly classified based on NLP, Statistical Methods and Machine Learning. A detailed review of some of the most important automatic slide generation techniques from academic articles is presented and a brief comparison among the discussed techniques is given. Including information on Generation of Slides Based on Inference of Underlying Semantic Structure of Articles, Generation of Slides from the LATEX Manuscript of an Article, Generation of Slides Using Natural Language Processing, Generation of Slides Using Text Summarization, Generation of Slides Using Web Mining, Generation of Slides Based on Machine Learning and also about comparisons of techniques and experimental results and analysis.

In paper[8], the author presented an approach It presented a novel unsupervised approach to the problem of multi-document summarization of scientific articles, in which the document collection is a list of papers cited together within the same source article, otherwise known as a co-citation. At the heart of the approach is a topic-based clustering of fragments extracted from each co-cited article and relevance ranking using a query generated from the context surrounding the cocited list of papers. This analysis enables the generation of an overview of common themes from the co-cited papers that relate to the con text in which the co-citation was found. We present a system called SciSumm that embodies this approach and apply it to the 2008 ACL Anthology. We evaluate this summarization system for relevant content selection using gold standard summaries prepared on principle-based guidelines. Evaluation with gold standard summaries demonstrates that our system performs better in content selection than an existing summarization system (MEAD). We present a detailed summary of our findings and discuss possible directions for future research.

In paper[9], the author presented an approach to the best of our knowledge, we are the first to use Deep Neural models to encode sentences and its context as new features in sentence ranking to build slides.) We combine regression with integer linear programming (ILP) to select salient sentences. Then, noun phrases are extracted from the selected sentences to build the first-level bullet points in slides. Here, we focus on the textual components. This approach can later be enhanced with visual effects, e.g., figures and tables, in order to make better more complex slides. Previously, a method was proposed to generate slides from documents by identifying important topics and then adding their related sentences to the summary.

In paper[10], the author presented an approach Creating presentation materials requires complex multimodal reasoning skills to summarize key concepts and arrange them in a logical and visually pleasing manner. Creating presentations is often a work of art. It requires skills to abstract complex concepts and conveys them in a concise. To tackle this task, we present a hierarchical recurrent sequence-to-sequence architecture that “reads” the input document and “summarizes” it into a structured slide deck. Despite this large body of work, there remain many tasks that have not been addressed, e.g., multimodal document generation. We extract sentence and figure embedding from an input document and project them to a shared embedding space so that the OP treats both textual and visual elements

In paper[11], the author presented an approach We paraphrase sentences before placing them on slides. This step is crucial because without it the text would be too verbose for a slide presentation. We design a learning objective that captures both the structural similarity and the content similarity between the ground-truth slides and the generated slides. That is, once our model generates a slide deck, we remove figures that have relevance scores lower than a threshold and add figures with scores higher than a threshold. We describe our approach for automatically generating presentation slides for scientific papers using deep neural networks. Such slides can help authors have a starting point for their slide generation process.

In paper[12], the author presented an approach a good slide deck should put text with relevant figures to make the presentation informative and attractive. A good design layout makes it easy to consume information presented in slides. We ablate different terms in the content similarity loss. The OP predicts the layout to decide where and how to put the extracted objects. We compare this with a template-based approach, which selects the current section title as the slide title and puts sentences and figures in the body line-by-line. We evaluate performance in a topic-dependent and independent fashion.

In paper[13], the author presented an approach the two major components of extractive summarization are sentence scoring and sentence selection. The goal is to keep salient sentences while excluding redundant information. Sentence scoring is usually converted to a regression problem. The scores depend on features extracted from the current sentence and its contextual sentences Extractive summarization techniques are applied to rank and select important sentences from the original document.

In paper[14], the author presented an approach although summarization has been well studied, the task of automatically generation of slides is relatively new. This paper uses neural network models to encode syntax, semantics, and context of sentences as features to automatically generate slide content. Our model then selects salient sentences with the constraint of a limit on summary length. Our method shows higher ROUGE scores compared with the introduced baselines which we interpret to mean that the slides generated have a high overlap with manually made slides.

In paper[15], the author presented an approach although the dataset is a collection of 1200 scientific papers and their corresponding presentation slides made by their authors. The dataset contains conference proceedings in computer science in PDF format. To extract the header metadata and content of the paper from PDF files, is used and information is transformed. In order to set the titles of the slides, the section of the first sentence in the slide is determined and then the heading of the section is borrowed from the paper as the title of the slide. The heading is truncated to the first 5 tokens.

SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements:

1. Operating System: The application can run on various operating systems. Common choices include Linux (e.g., Ubuntu), Windows Server, or containerized solutions like Docker.
2. Python: Ensure that Python is installed on the server. The provided code uses Python with Flask, gensim, pdfplumber, and pptx libraries.
3. Flask: The Flask web framework is used for creating the web application.
4. Gensim: Gensim is used for text summarization.
5. pdfplumber: The pdfplumber library is used for extracting text from PDF files.
6. pptx: The pptx library is used for generating PowerPoint presentations.
7. Tempfile: The Python tempfile module is used for creating temporary files.
8. Random: The random module is used for adding randomness to the selection of content.
9. Web Server: You need a web server to host the Flask application. Common choices include Gunicorn, uWSGI, or you can use Flask's built-in development server for testing (not recommended for production).
10. Web Browser: Users will access the application via a web browser.
11. Network Access: Ensure the server has internet access if any libraries or resources need to be fetched from the internet.
12. Dependencies: Install all necessary Python dependencies using a virtual environment to isolate the application's requirements.

Hardware Requirements:

1. Server or Hosting: A server or hosting environment to run the Flask application. This can be a physical server or a cloud-based virtual machine (e.g., AWS, GCP, Azure).
2. CPU: Intel Core i5-11400 2.6 GHz Six-Core LGA 1200 Processor into your computer system.
3. RAM: 4GB of RAM or more is recommended.
4. Storage: 512GB M.2 NVMe™ PCIe® 3.0 SSD

REFERENCES

1.Koning, B. (2022). Extracting Sections From PDF-Formatted CTI Reports (Bachelor's thesis, University of Twente).

2.Vyshnavi, V. R., & Malik, A. (2019). Efficient Way of Web Development Using Python and Flask. Int. J. Recent Res. Asp, 6(2), 16-19.

3.Srinivasa-Desikan, B. (2018). Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim , Spacy, and Keras. Packt Publishing Ltd.

4.Řehůřek, R., & Sojka, P. (2011). Gensim—statistical semantics in python. Retrieved from genism. org.

5.SLIDEGen: Approach to automatic Slides GenerationMiss. Autade Dhanshri P 1, Prof. Raut S.Y2

6.Girardi, E. A. (2022). Automatic slide generation from scientific papers based on multimodal learning (Doctoral dissertation, Politecnico di Torino).

7. A Comparison on Techniques for Automatic Generation of Presentation Slides Biju P. Dais P G Scholar Department of Computer Science & Engineering College of Engineering,
Perumon(CUSAT), Kerala, India Smitha C.S. Assistant Professor in CSE Department of Computer Science & Engineering College of Engineering, Perumon(CUSAT), Kerala,
India.

8.Nitin Agarwal, Ravi Shankar Reddy, Kiran Gvr, and Carolyn Penstein Rosé. 2011. Towards Multi-Document Summarization of Scientific Articles:Making Interesting Comparisons with SciSumm. In Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages, pages 8–15, Portland, Oregon. Association for Computational Linguistics.

9.Y. Yasumura, M. Takeichi, and K. Nitta, “A support system for making presentation slides,” Trans. Japanese Soc. Artif. Intell., vol. 18, pp. 212–220, 2003.

10.S. M. A. Masum and M. Ishizuka, “Making topic specific report and multimodal presentation automatically by mining the web resources,”in Proc. IEEE/WIC/ACM Int. Conf. Web Intell., 2006, pp. 240–246.

11.N. Agarwal, K. Gvr, R. S. Reddy, and C. P. Ros_e, “Towards multidocument summarization of scientific articles: Making interesting comparisons with SciSumm,” in Proc. Workshop Autom. Summarization Different Genres, Media, Lang., 2011, pp. 8–15. [6] A. Abu-Jbara and D. Radev, “Coherent citation-based summarization of scientific papers,” in Proc. 49th Annu. Meeting Assoc. Comput.Linguistics: Human Lang. Technol.- Volume 1, 2011,pp. 500–509.

12. K. Woodsend and M. Lapata, “Multiple aspect summarization using integer linear programming,” in Proc. Joint Conf. Empirical Methods Nat. Lang. Process. Comput. Nat. Lang. Learn., 2012,pp. 233–243.

13.P. B. Baxendale, “Machine-made index for technical literature: an experiment,” IBM J. Res. Develop., vol. 2, no. 4, pp. 354–361,1958.

14. Sujian Li, You Ouyang, Wei Wang, and Bin Sun. 2007. Multi-document summarization using support vector regression. In Proceedings of DUC. Citeseer.

15.Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out (2004)