BIBTEX AI: LLM BIBTEX PRESENTATION TOOL

Thesis submitted in partial fulfillment of the requirements for the award of the degree of ***Master of Computer Applications*** *of the* ***APJ Abdul Kalam Technological University***

submitted by

ABHISHEK A
(TCR23MCA-2001)

DEPARTMENT OF COMPUTER APPLICATIONS
GOVERNMENT ENGINEERING COLLEGE
THRISSUR - 680009

APRIL 25

DEPARTMENT OF COMPUTER APPLICATIONS
GOVERNMENT ENGINEERING COLLEGE, THRISSUR
THRISSUR, KERALA STATE, PIN 680009

CERTIFICATE

This is to certify that the mini project titled **"BIBTEX AI: LLM BIBTEX PRESENTATION TOOL "** *is a bonafide work done by* **ABHISHEK.A (TCR23MCA-2001)** *under my supervision and guidance, and is submitted in Apr 2025 in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications from APJ Abdul Kalam Technological University(KTU).*

Prof. Raseek C	Dr. Sminesh C N
Project Guide	**Project Coordinator and HOD**

Place : THRISSUR
Date : 04-04-2025

DECLARATION

I hereby declare that the mini project named, **BIBTEX AI: LLM BIBTEX PRESENTATION TOOL**, is my own work and that, to the best of my knowledge and belief, it contains no material previously published by another person nor material which has been accepted for the award of any other degree or course of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place : THRISSUR	Signature
Date : 04-04-2025	**ABHISHEK.A (TCR23MCA-2001)**

ACKNOWLEDGEMENT

I would like to thank Department of Computer Applications, GEC Thrissur, for giving me this opportunity to pursue this project and successfully complete it.

I am highly indebted to my project guide **Prof. Raseek C** for his committed guidance, valuable suggestions and constructive criticisms. I thank him for the precious time that he invested throughout the work and for backing me when there were downs on the project.

I would like to express my heart-felt gratitude to my Project Coordinator and HOD **Dr. Sminesh C N** , for his guidance as well as for providing necessary information regarding the project and also for his support in completing the project.

I sincerely thank all other faculties Department of Computer Applications, for guiding through the processes involved in the project.

ABSTRACT

In the era of advanced automation and artificial intelligence, large language models (LLMs) have unlocked new possibilities for simplifying complex tasks. This project focuses on developing an intelligent tool to enhance productivity and creativity for students, researchers, and professionals by streamlining the process of creating professional academic presentations and reports, particularly in Overleaf Beamer, a widely used LaTeX-based platform. The system leverages an innovative multi-agent workflow, where distinct agents collaborate to perform specialized tasks, such as generating structured content, comparing research papers, integrating citations, and ensuring LaTeX compliance.

The project aims to make academic document generation more efficient and user-friendly by enabling users to upload research papers, customize templates, and interactively refine the generated content. With features like automated reference integration and the tool reduces the challenges associated with LaTeX and academic formatting. This system provides a seamless and intelligent solution for structured content generation, empowering users to focus more on research and knowledge sharing rather than technical complexities.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Creating academic presentations and reports is a key part of modern education and research. Overleaf Beamer, a LaTeX-based platform, is widely recognized for producing professional and well structured slides. However, while LaTeX offers great flexibility and precision, its complexity can be overwhelming especially for students and researchers who are new to the platform. The steep learning curve often reduces productivity and slows down the creative process.

Artificial intelligence, particularly Large Language Models (LLMs), is now making it easier to handle these challenges. LLMs are trained to understand and generate human-like text, making them powerful tools for automating tasks such as content creation, citation management, and document formatting. While previous tools have used LLMs for converting text into presentations, many are limited in scope and lack dynamic interaction.

This is where the concept of *multi-agent* systems comes in. These systems use multiple specialized agents that collaborate like a team of people to perform different tasks. Each agent contributes its part, shares results with others, and refines the output together. This cooperative method allows for better performance, multitasking, and more accurate results.

A new approach called BibTex-AI introduces an advanced tool that simplifies the process of creating academic presentations and reports. It works closely with LaTeX and uses the capabilities of LLMs to make content generation easier especially for Beamer presentations. The system relies on a multi-agent workflow to analyze documents, manage references, and ensure proper formatting, making it much easier for users to focus on their ideas rather than formatting issues.

The goal of this project is to offer a user friendly solution that helps even those unfamiliar with LaTeX. It supports users in organizing their thoughts clearly and presenting their work effectively. With BibTex-AI, the limitations of LaTeX no longer need to hold back students or researchers. This project promotes a smarter, more collaborative way of generating academic content, pointing toward the future of automation and intelligent document creation.

CHAPTER 2
LITERATURE REVIEW

Cutting-edge developments in Artificial Intelligence (AI), Large Language Models (LLMs) and their capacity to automate complex content-generation tasks have literally become a watershed in the expansive automation of many complex tasks. The future of agentic workflows-where numerous agents pool their expertise to reach one common objective-has come to the fore in recent years. This model, which is wholly owed to multiagent systems, has been very effective in decision-making, task automation, and collaborative problem-solving. This literature review sets the foundation for the project by highlighting the various research articles where LLMs, agentic workflows, and multiagent systems have been explored for academic and professional content creation. These studies generally highlight the potential for collaboration driven by AI and the attendant bones of contention and limitations for implementing them, which in turn guide the development of this project.

The relevance and significance of the LATEX System to the generation of tagged PDF is based on the most fundamental property of a LATEX document: that it is a tagged document. Specifically, a LATEX file is a ’free-form’ text file specifying the structure of a document in terms of markup tags, and representing its textual content as unformatted character data. A LATEX file does need to include any explicit formatting data (no font data nor layout data). All such non-structural data that is required to generate a formatted and typeset document (usually a PDF file) is usually kept independent of the document itself, in LATEX package files and class files.[1]

Ayush Thakur[2] examined the complex role of prompting for Large Language Models (LLMs) and how it affects their performance and abilities. In this thorough analysis, we have explored the nature of prompting, highlighting its relevance and uncovering the techniques needed to use it efficiently. We began by explaining the key ideas of prompting and how it helps LLMs work better, building a basic knowledge. Then we analyzed the structure of a prompt, identifying the important elements for making good prompts, and also pointing out the frequent mistakes to avoid along the way. The art of prompting opens up exciting prospects for the future of AI. As AI keeps advancing, we can expect to see new trends and breakthrough applications of sophisticated prompting methods. Prompting plays a crucial role in guiding the direction of AI development, creating a future where these models overcome current challenges.

Sanka Rasnayaka et al.[3] carried out research which finds that LLMs are hugely useful at the outset of software development projects, boosting productivity in mundane tasks such as debugging. Yet, the use of LLMs differed according to the coding abilities and experience with AI generators of the students, pointing to a learning curve in the effective utilization of these tools. We also found no substantial disparity in the quality and accuracy of software developed by teams with high and zero AI usage.

Qinjin Jia et al.[4] suggested that AI-powered automated feedback systems based on LLM will be an important aspect of the future AI-based learning system. But issues related to hallucination hinder the integration of feedback systems into real classroom environments. In this paper, we have explored student project report feedback from both data-driven and prompt-driven systems. The outcome shows both systems generate a significant number of intrinsic and extrinsic hallucinations. Further, the fine-tuning of ChatGPT better correspondence with human judgment in assessing hallucinations, but more powerful approaches remain to be discovered.

Hao Zheng et al.[5] introduced PPTAgent, which models presentation generation as a two-step presentation editing process done through the capabilities of LLMs to comprehend and produce code. It utilized the text feature and layout patterns to structurally classify slides into various functional categories. Our experiments on data from various domains have proven the effectiveness of our method.

Shuofei Qiao et al.[6] proposed a unified agentic workflow generation benchmark with miscellaneous scenarios and intricate graph-structured workflows. To precisely assess the workflow generation capability of LLM agents, we further present WORFEVAL, which utilizes quantitative algorithms to evaluate both the linear and graph workflows. This research introduce WORFBENCH, a unified agentic workflow generation benchmark with miscellaneous scenarios and intricate graph-structured workflows. Through comprehensive experiments across various kinds of LLMs, there large performance gaps between the traditional linear structured and complex graph-structured workflow generation.WORFBENCH provides a diverse benchmark for testing workflow generation, while WORFEVAL introduces a systematic evaluation method using algorithms for subsequence and subgraph matching to assess the workflow generation capabilities of LLMs.

Jiayi Zhang et al.[7] conducted a study on establishing a foundational structure for future research. AFLOW has leveraged Monte Carlo Tree Search and code-represented workflows to navigate the vast search space of possible workflows efficiently. Our experiments across six benchmarks demonstrate the effectiveness of AFLOW, which has outperformed manually designed methods and existing automated optimization approaches. Ablation studies have shown that AFLOW can autonomously discover effective structures, even without predefined operators.

Zijun Liu et al.[8] presents an architecture called Dynamic LLM-Powered Agent Network (DyLAN) to facilitate collaboration between dynamic teams of agents on complex tasks. DyLAN operates in a two-stage paradigm, allowing agents to collaborate in a dynamic structure with reformation of agent teams. In ”Team Optimization” phase, the agent selection algorithm using an unsupervised measure called Agent Importance Score, identifies best contributory agents in a principled manner for collaboration in ”Task Solving”. DyLAN presents a dynamic agent collaboration framework that maximizes task-solving by choosing the most contributory agents according to the Agent Importance Score. DyLAN provides significant improvements in accuracy and efficiency compared to static frameworks, paving the path for more adaptive and scalable LLM-driven systems.

The rapid advancements in Artificial Intelligence (AI) and the emergence of Large Language Models (LLMs) have led to groundbreaking innovations in automating complex tasks, particularly in content creation and structured document generation[15]. In recent years, the concept of agentic workflows, where multiple specialized agents collaborate to achieve a unified goal, has gained significant attention. This approach, inspired by multi-agent systems, has been effective in domains such as decision-making, task automation, and collaborative problem-solving. To provide a foundation for this project, this literature review explores key research contributions that highlight the applications of LLMs, agentic workflows, and multi-agent systems in academic and professional content creation. These studies not only focus on the possibilities of the collaboration, but on the challenges and limitations associated with AI-driven approaches and guide further development of the project.

CHAPTER 3
ENVIRONMENTAL STUDY

3.1 System Configuration

System configuration describe the hardware and software requirement of the system for development

3.1.1 Hardware Requirements

- Computer or Laptop with internet connectivity
- Storage capacity of at least 256GB
- Minimum 8GB RAM for efficient processing
- Multi-core processor (Intel i5 or AMD equivalent and above)
- External storage device (optional, for data backups)
- High-resolution monitor (optional, for improved content editing experience)

3.1.2 Software Requirements

3.1.2.1 Functional Requirements

- **User Input and Prompt Handling** The system should allow users to input specific Research Papers for desired tasks and the system will create prompt based on it and send it to LLM. which would be used as the starting point for content generation.
- **Support for Multiple Research Paper Uploads**

  - The system should enable users to upload multiple research papers for analysis.
  - Users should have options to specify their desired presentation type:
    * Compare and synthesize insights from the uploaded research papers.
    * Create a presentation based on the summary of all the uploaded papers.

- **Agent-Oriented Task Allocation** The system shall employ a multi-agent architecture with agents specializing in different functionalities:
  - **Content Extraction Agent:** Responsible for Extracting contents from the input of the user.
  - **Prompt Agent:** Prompt Agent generates a well strcutured prompt and sends it to the LLM the best output to be produced by the LLM.
  - **Citation Agent:** Fetches citations from citation input research papers and integrates them into BibTeX format.
  - **Report Generation Agent:** Produces detailed reports for the users in addition to slides for presentations of their topics.
- **Inter-Agent Communication** The agents should be able to exchange intermediate results with each other to refine tasks collaboratively. This will help in synchronizing and producing cohesive outputs.
- **Output Delivery**
  - Deliver structured BibTeX files optimized for use in Overleaf Beamer.
  - Offer PDFs and editable LaTeX files as output formats, making the system versatile and user-friendly.

3.1.2.2 Non-Functional Requirements

- Performance
- Usability
- Compatibility
- Scalability

3.1.3 Software Development Requirements

- **Operating System:** Windows 10 or later.
- **Front End:** Python 3.11.4 (Streamlit for user interface development)
- **Back End:** Python (FastAPI for API development), with integration of LLM APIs
- **IDE Used:** VS Code 1.81 or later
- **Additional Software:** Overleaf for LaTeX document preparation.
- **Libraries and Frameworks:** Pandas, NumPy, PyLaTeX, and LangChain for LLM-based agent management

3.2 Software Specifications

3.2.1 Operating System

Windows 10 or later, macOS, or Linux is the recommended operating system for running the software. The software should be cross-platform compatible and function well on these platforms.

3.2.2 Programming Languages

- **Python 3.11**: Python will be in charge of most of the backend development activities such as providing communication between agents, processing user inputs, and managing interactions with external APIs.

The system will utilize Python libraries in agent communication, generation of content, and LaTeX formatting.

- **LaTeX**: LaTeX is used in the generation of formatted presentations and reports, especially while preparing Beamer-based presentations. In this case, the system is expected to be outputting to LaTeX formats that are both Overleaf-compatible and other similar LaTeX editors.

3.2.3 Frameworks and Libraries

- **TensorFlow or PyTorch**: If generating content or text synthesis involves machine learning models, one of these will be used to build and deploy that kind of model. It could be helpful in building summaries or content out of research papers.
- **LaTeX Libraries**: Specific LaTeX libraries will be used to ensure proper formatting, error checking, and LaTeX syntax validation. These libraries will be used to automate the generation of well-structured and error-free presentations and reports.

3.2.4 Integrated Development Environments (IDEs)

Visual Studio Code 1.81: This will be used for writing, debugging, and testing Python code. It supports Python extensions, which will help manage the development of the backend system.

3.2.5 Citation Management

The citations will be retrieved automatically and used to build structured LaTeX bibliographies, thus streamlining the process of citation for the user.

3.2.6 Other Software

- **Overleaf**: Overleaf is used for editing LaTeX documents in real time, collaborating, and generating PDF outputs. It will be integrated with the system to enable seamless work on the contents that are generated as LaTeX.

- **Beamer**: Beamer will be used for creating professional presentations as LaTeX. The generated slides through the Beamer package will be automated by the system so that the users can work directly on their presentation from the input data.

CHAPTER 4
SYSTEM ANALYSIS

The practice of performing a systems analysis on a business situation in an effort to design a system solution to an issue or develop enhancements to that context is referred to as systems improvement. Prior to the initiation of developing any system, the project proposal is drawn up by the prospective users and/or systems analysts and presented to the relevant managerial hierarchy in the company.

The route to questioning a business scenario in an effort to enhance it via enhanced strategies and processes is integral to a quality system inquiry. System analysis is done to examine a system or its parts in an effort to identify its objectives. It is a problem-solving technique that improves the system and ensures all its components work well to do what they are designed for.

4.1 Requirements Analysis

Requirements analysis, or also called requirements engineering, is how requirements are gathered, analyzed, and documented regarding end-users' needs for a new or changing system. They must be concrete, quantifiable, and have value to the purpose of the project. When done in the creation of software, they lay the basis for implementation and system design.

The analysis involves understanding user needs and expectations while ensuring that the requirements are practical and achievable within the project's scope. This includes identifying both functional and non-functional requirements, such as system performance, usability, and compatibility. The focus is on providing a seamless experience for end-users by addressing their primary needs and ensuring the system operates efficiently across various environments.

A comprehensive approach to requirements analysis helps in mitigating risks, improving system quality, and ensuring that the final product aligns with user expectations. Proper documentation of these requirements serves as a guide for subsequent stages of system design, development, and testing.

4.2 Existing Systems

The current academic report and presentation-generation systems mainly focus on using very manual processes or have significantly limited automation levels. A very common set-up for such documentation is by employing text editors like Overleaf as dedicated LaTeX interfaces[9]. The set-ups do very well to take care of full LaTeX syntax support with templates but the user still requires a pretty sound knowledge of the LaTeX commands and their typesetting. Another advantage is that the process of citation management usually works separately. So, it asks for users to collate and format their references manually. Hence, this segregation between content generation, citation management, and formatting contributes to wasting huge amounts of users' time and effort, more so when one is a LaTeX and academic writing novitiate. Currently, existing systems are not taking advantage of the sophisticated technologies of large language models or multi-agent systems in simplifying and streamlining these tasks, which means a gap exists in user productivity and accessibility.

4.3 Limitations of existing systems

- Manual content creation.
- Steep learning curve for LaTeX-based platforms.
- Fragmented workflow requiring multiple tools.
- Limited automation for content generation and formatting.
- Error-prone processes due to manual LaTeX coding.

- Lack of advanced technologies like large language models.
- No real-time integration with citation databases.
- Inconsistent output quality depending on user expertise.
- Limited personalization options for templates and layouts.
- Platform dependency and restricted accessibility.

4.4 Proposed System

The proposed system introduces a smart, multi-agent solution with large language models (LLMs) to enable streamlined creation of academic presentations and reports. It automatically generates structured content from a research paper or user input and facilitates smooth integration into LaTeX-based environments like Overleaf. The system, leveraging a collaborative multi-agent workflow, performs specialized tasks like extracting research highlights, formatting content, ensuring LaTeX compliance, and real-time management of citations through BibTeX entries. The system will also include error detection and correction for LaTeX syntax to generate error-free documents compatible with Beamer templates. Users will also be able to interactively refine and customize the generated content, including templates and layouts, for a personalized experience. The proposed solution is to be efficient on multiple platforms: Windows, macOS, and Linux. This makes the creation of academic content more efficient and user-friendly for students, researchers, and professionals.

4.5 Advantages of Proposed System

- Automates content generation for academic presentations and reports.
- Streamlines LaTeX formatting, ensuring compliance with Beamer templates.
- Real-time citation management

- Error detection and correction for LaTeX syntax, ensuring error-free document compilation.
- Multi-agent workflow for task specialization, improving efficiency and accuracy.
- Customizable templates and layouts to suit user preferences.
- Enhances productivity and creativity for students, researchers, and professionals.

4.6 Input and output

4.6.1 Input

The inputs for the system are as follows:

- **Format PDF** : The Format PDF is used so that the system gets a clear idea of in which format we want our final output in.
  - IEEE paper as format
  - Beamer slides as format
- **Research Papers**: Users can upload multiple research papers, which the system will analyze to extract key insights, compare findings, and generate relevant content for the presentation or report.

4.6.2 Output

The primary output generated by the system is:

- **LaTeX Beamer Presentation**: The system produces a professionally formatted Beamer presentation in LaTeX, ready for use in Overleaf or similar LaTeX editors. The presentation is based on the analysis of the user-provided research papers and prompts, and is structured according to the generated content and customized templates.

- **Overleaf LaTeX Report:** In addition to the LaTeX-based presentation slides, the system also generates a detailed LaTeX report based on the research papers and user input. The report is structured in a proper way, with proper citation management, section organization, and academic standards. The system integrates relevant references, automatically generates a bibliography using BibTeX, and ensures LaTeX compliance for seamless compilation in Overleaf or other LaTeX environments.

4.7 Feasibility Study

An important level container shape of the whole system analysis and design process is a feasibility study. The problem definition is then grouped for the analysis. Determine whether the task is feasible before attempting it. A valid model of the structure is built by the specialist after an acknowledgement issue specification has been created. The search for options is carefully examined. The feasibility study consists of 3 parts.

4.7.1 Technical Feasibility

The technical feasibility of the proposed system has been thoroughly assessed, and the project can be successfully implemented using available technologies. The backend will be developed using Python 3.11, agent communication, data processing, and interaction with external APIs. Due to the rich ecosystem of libraries provided by Python, including Requests and LaTeX packages, the functionality of the system will be implemented very efficiently. For its own part, LaTeX templates through Beamer come up as providing the reliable integration by itself into these tools to render latex-based presentation that would greatly allow feasibility within these integration points.

Content generating along with producing synthetic text involves application of appropriate models like those involving TensorFlow, or PyTorch which can sustain intelligent agent programming development within systems. Real-time citation management with the use of Python libraries, that enables seamless retrieval and formatting of citations as BibTeX entries. The system will also be cross-platform, thus running well on Windows, macOS, and Linux platforms, and again both Python and LaTeX are generally cross-platform. Even the development environment with Visual Studio Code will be technically feasible for full support with Python, LaTeX, and version control. The integration of LaTeX for the generation of Beamer presentations is entirely feasible given the widespread use of LaTeX in academic and professional settings. Overall, the project is technically sound and achievable with current technologies.

4.7.2 Operational Feasibility

The operational feasibility of the proposed system is evaluated relative to the extent of its alignment with end-users' needs and its integrating into their workflow. The system is designed to automate professional academic presentations and reports, thereby not requiring much human effort in formatting and citation management. This will considerably boost productivity for students, researchers, and professionals through streamlined LaTeX document preparation. It will make use of well-known technologies like Python, LaTeX, and Beamer, which ensures that the user is already familiar with these tools or easily gets along with them. The system is also integrated with citation management to allow the users to fetch references without much hassle. The system is interactive, and thus, it can be customized. Users can customize the output according to their preferences. The cross-platform compatibility of the system will enable users from different operating systems, such as Windows, macOS, and Linux, to use the system effectively. Training users on how to interact with the system is relatively simple, as the user interface is designed to be intuitive and user-friendly. Therefore, the system is operationally viable and provides concrete benefits that accrue through its adoption.

4.7.3 Economical Feasibility

The economic feasibility of the proposed system has been assessed by considering the costs involved in its development, deployment, and maintenance as well as the potential benefits it offers to the users. The project uses open-source technologies like Python, LaTeX, and TensorFlow, which reduces the cost of licensing proprietary software significantly. In addition, the use of cloud-based tools such as Overleaf minimizes the need for expensive local infrastructure. The automation of other time-consuming tasks, such as content generation, formatting, and citation management, can add up to considerable time savings for the users, mainly researchers, students, and academics who quite often deal with LaTeX-based documents. This could mean a higher throughput of presentations and reports, with possible cost savings on labor for the purposes of organizations that require frequent creation of academic documents. The cross-platform nature of the system also allows a user base regardless of operating system, meaning they do not necessarily have to waste money on upgraded hardware. On account of improved productivity and workflow, it allows for high returns on investment as a whole both in academia and the professional environment, and is indeed economically feasible considering the high worth it gives without too much financial burden on the expense of its construction and operational run.

CHAPTER 5
SYSTEM DESIGN

5.1 Application Architecture

Fig. 5.1 shows the architecture of BIBTEX AI. This model represents the main functionalities of the BIBTEX AI.

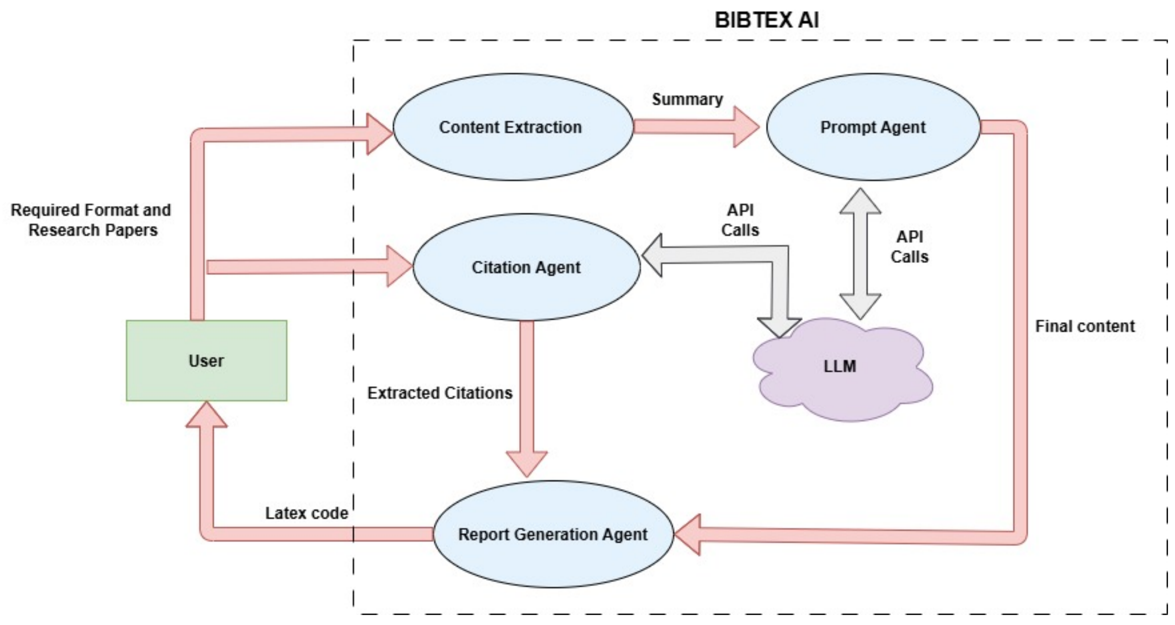


Fig. 5.1: System Architecture

The main activities of BIBTEX AI are:

- **User Input and System Initialization:**
  - The User starts the process by providing:
    - Research papers in PDF, text, or structured format.
    - Format specifications required (for example, IEEE, ACM, Beamer slides).

  - The system (BibTeX AI) initializes by validating the input and getting agents ready for processing.
  - The user's input is passed to the Prompt Generation Agent for further processing.

- **Prompt Generation:**
  - This process interprets the user's request and generates structured prompts for further processing.
  - The generated prompts include:
    - Key questions to extract information from research papers.
    - Formatting instructions about the report or presentation.
    - Specific requests for content; for example, summarization, comparison, and citation management.
  - The prompts are sent to:
    - The LLM through API calls for further processing of text content.
    - The Keypoint Extraction Agent for deeper analysis of the research paper.
- **Content Extraction:**
  - The Content Extraction Agent accepts the research papers and extracts the contant from the input and performs the following tasks:
    - Identifies important parts, including abstract, introduction, methodology, and conclusion.
    - Extracts keypoints based on key topics, such as problem statements, findings, and contributions.
  - The resulting keypoints are transferred to the Prompt Agent.
- **Citation Handling:**

  - The Citation Agent checks all references for proper citations in the relevant format.
  - Tasks performed by the agent:
    * Retrieves names of authors, paper titles, publication information, and DOI links.
    * Converts citations to BibTeX so they can easily be included within LaTeX documents.
  - The enriched citations are then handed to the Report Generation Agent.

- **Report or Beamer presentation in LaTeX:**
  - The Report Generation Agent is responsible for assembling the final structured document.
  - It performs the following tasks:
    * Organizes the extracted and cited content into a LaTeX (Beamer) template or LaTex Report.
    * Structures sections such as title, abstract, introduction, methodology, results, discussion, and references.
    * Ensures LaTeX syntax compliance by checking for formatting errors.
    * Inserts proper BibTeX citations into the bibliography section.
  - The final LaTeX code is generated and sent back to the user.

- **User Receives Final Output:**
  - The user receives the output as a LaTeX (.tex) file, containing:
    * Fully formatted research paper or presentation.
    * Properly cited references in BibTeX format.
    * Structured content ready for Overleaf or other LaTeX editors.
  - The user can further modify, compile, and export the document as needed.

- **Key System Interactions:**
  - **LLM API Calls:** Used for generating summaries, refining research comparisons, and structuring content[16].
  - **Agent Communication:** Agents work collectively to share the extracted content, refining the output before final generation.
  - **Automated Citation Handling:** Ensures references are correct, formatted appropriately, and put into the report correctly.

5.2 Data Flow Diagram

The figure 5.2 - 5.3. is the Data flow diagram of BIBTEX AI. It shows the overall systems Data flow and the process.

5.2.1 DFD LEVEL0

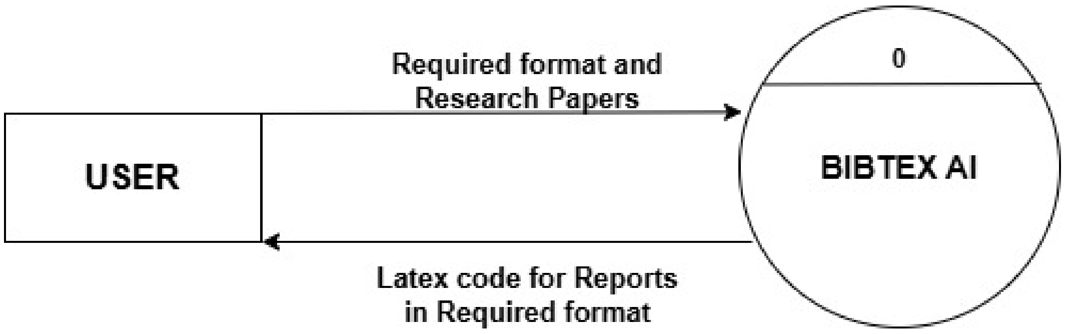

Required format and Research Papers
0
USER
BIBTEX AI
Latex code for Reports in Required format

Fig. 5.2: Data Flow Diagram Level 0

5.2.2 DFD LEVEL 1

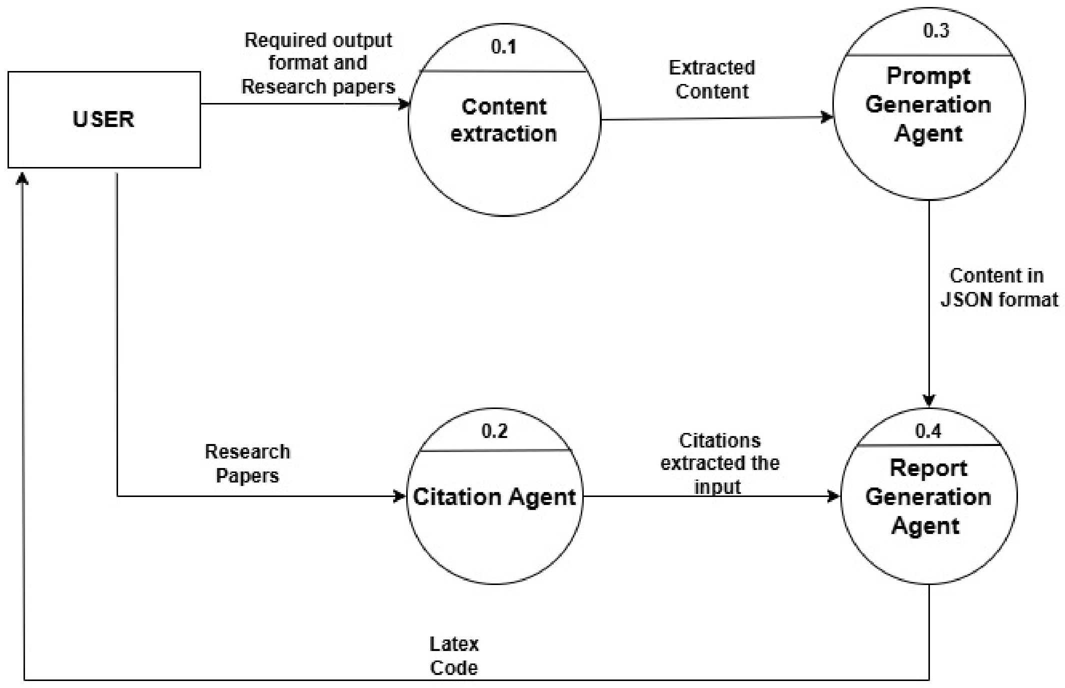


Fig. 5.3: Data Flow Diagram Level 1

5.3 List of Modules

BIBTEX AI mainly consist of eight modules, They are:

- User Interface Module
- Prompt Generation Module
- Content Extraction Module
- Citation Handling Module
- Report Generation Module
- LLM Integration Module
- System Communication Module

5.3.1 User Interface Module

This module serves as the primary interaction point between the user and the system. It allows users to upload research papers in different formats, specify formatting preferences, and configure processing options. Once the LaTeX document is generated, the module provides a preview and options to download or edit the output before finalizing the document.

5.3.2 Prompt Generation Module

The prompt generation module interprets the user′s input and constructs well-defined prompts for the system. These prompts help define the scope of content extraction, including identifying key topics, structuring the document, and setting formatting preferences. This module ensures that the generated prompts are well-structured for efficient processing by the system.

5.3.3 Content Extraction Module

This module extracts important sections from the provided research papers, such as the abstract, introduction, methodology, and conclusion. It identifies key arguments, findings, and contributions from the text, ensuring that only relevant and essential content is used in the report generation process. This step helps in refining the research data before further processing.

5.3.4 Citation Handling Module

This module is responsible for managing and formatting citations based on the references present in the submitted research papers. It extracts citation data from the provided documents, ensuring proper reference structuring in BibTeX format. The module ensures that citations are correctly formatted and placed in the appropriate sections within the final LaTeX document.

5.3.5 Report Generation Module

The report generation module compiles all processed content into a structured LaTeX document. It organizes extracted and cited content into predefined LaTeX templates, ensuring proper document structure, sectioning, and formatting. Additionally, it checks for LaTeX syntax errors and ensures proper bibliography integration before finalizing the document.

5.3.6 LLM Integration Module

This module connects with the Large Language Models (LLMs) via API calls for refinement of content, generation of well-structured summaries, and helping in comparison for research. It helps to increase the clarity and coherence of generated text and make it more worthy of high-quality academic writing.

5.3.7 System Communication Module

The system communication module manages interactions between different agents in BibTeX AI. It ensures seamless data exchange, manages dependencies between processing steps, and optimizes workflow execution. This module is crucial for maintaining the efficiency and scalability of the system.

CHAPTER 6
SYSTEM IMPLEMENTATION

6.1 Overview

The BibTeX AI system was implemented as a multi-agent application for generating academic reports and presentations in LaTeX format using input research papers and user-defined output formats. The system integrates an LLM (via API), various agents for document processing, and a backend pipeline for content extraction, comparison, citation generation, and LaTeX formatting. The implementation process was divided into multiple phases, involving component-wise development, integration, and iterative testing.

6.2 LLM Interface Integration

The implementation began with integrating a Language Model API (initially Mistral, later exploring DeepSeek) to facilitate prompt-based content generation. This included setting up authorization headers, handling JSON payloads, and managing token limits. The LLM Interface was designed as a reusable component to interact with different agents through structured prompts.

6.3 Input Handling and PDF Extraction

6.3.1 Input Handler

The Input Handler module was developed to accept two primary inputs from the user:

- Research Papers in PDF format.

- A required format PDF (IEEE report or Beamer presentation).

The handler validates file types and structures the inputs to be passed through the pipeline.

6.3.2 PDF Extractor

Using Python libraries like `PyMuPDF (fitz)` and `pdfminer.six`, the PDF Extractor module was implemented to parse and extract readable content from uploaded PDFs. It converts raw documents into tokenizable text, preserving section headers and references wherever possible.

6.4 Pipeline Development and Testing

A modular pipeline was built to pass extracted input through a sequence of agents:

1. Prompt Generation Agent
2. Content Extraction Agent
3. Citation Agent
4. Report Generation Agent

The initial pipeline was created and tested, during which several debugging iterations were performed to fix flow errors, improve formatting consistency, and validate outputs.

6.5 Agent Development and Integration

6.5.1 Prompt Generation and Report Agent

The Prompt Generation Agent was the first agent developed. It interprets the user's input and generates a structured prompt for the LLM, customized according to whether the desired output is a research report or a Beamer presentation. The Report Generation Agent formats the final LaTeX document based on the processed content and template type.

6.5.2 Citation Agent

The Citation Agent was implemented to retrieve and format references in BibTeX using APIs such as CrossRef and Google Scholar. The JSON response parsing was refined to handle edge cases in publication metadata extraction.

6.5.3 Integration and Testing

After individual agent development, integration into the main pipeline was completed, followed by iterative testing and content improvements. Output quality was improved in terms of coherence, LaTeX structure, and citation placement.

6.6 PPT (Beamer) Generation

A key feature added was PPT generation using LaTeX Beamer. The system dynamically generates slides from extracted keypoints and formatted content. The Beamer template was customized to handle various content structures including title slides, bullet points, and reference slides.

6.7 Testing and Review Phases

Several rounds of testing were conducted:

- Pipeline testing with different paper inputs
- Debugging and adjustments after feedback (citation agent fixes, JSON parsing)

Final testing confirmed that the system could generate both reports and presentations from raw research content, preserving academic structure and formatting standards.

6.8 Tools and Libraries Used

- **Python 3.11** – Core development language for backend processing and agent orchestration.

- **Requests** – For interacting with external APIs (LLM, citation sources).
- **PyMuPDF (fitz), pdfminer.six** – For extracting text from PDF files.
- **LangChain** – To manage prompt chains and context-aware agent design.
- **LaTeX and Beamer** – Document formatting and professional presentation generation.

6.9 Summary

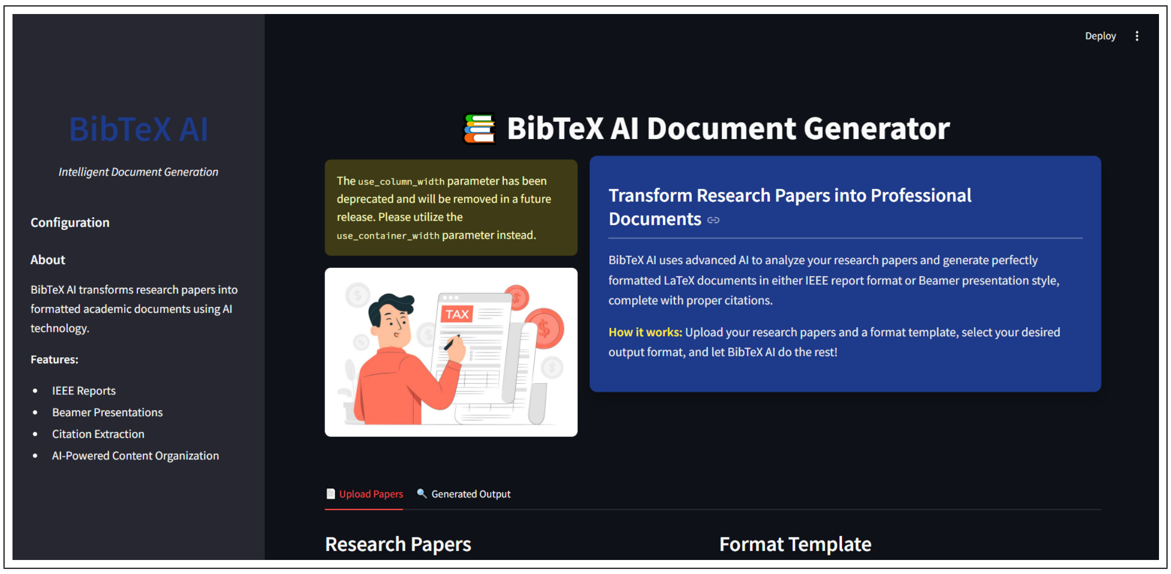
The implementation phase successfully transitioned from basic component development to an integrated, intelligent academic content generation tool. Each module was iteratively tested and refined to ensure modularity, scalability, and user-centric output quality.

CHAPTER 7
RESULTS AND DISCUSSION

7.1 Overview

The proposed system, **BibTeX AI**, was developed to automate the generation of LaTeX formatted academic content (reports and presentations) from user uploaded research papers. This chapter presents the visual output of the system and discusses its functionality across different formats. The results are based on interactions with both IEEE-style paper formatting and Beamer presentation formatting. Screenshots of the User Interface (UI), input formats, and generated output are presented for a comprehensive understanding.

7.2 User Interface



About

BibTeX AI transforms research papers into formatted academic documents using AI technology.

Features:

- IEEE Reports
- Beamer Presentations
- Citation Extraction
- AI-Powered Content Organization

BibTeX AI Document Generator

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Transform Research Papers into Professional Documents

BibTeX AI uses advanced AI to analyze your research papers and generate perfectly formatted LaTeX documents in either IEEE report format or Beamer presentation style, complete with proper citations.

How it works: Upload your research papers and a format template, select your desired output format, and let BibTeX AI do the rest!

Research Papers

Format Template

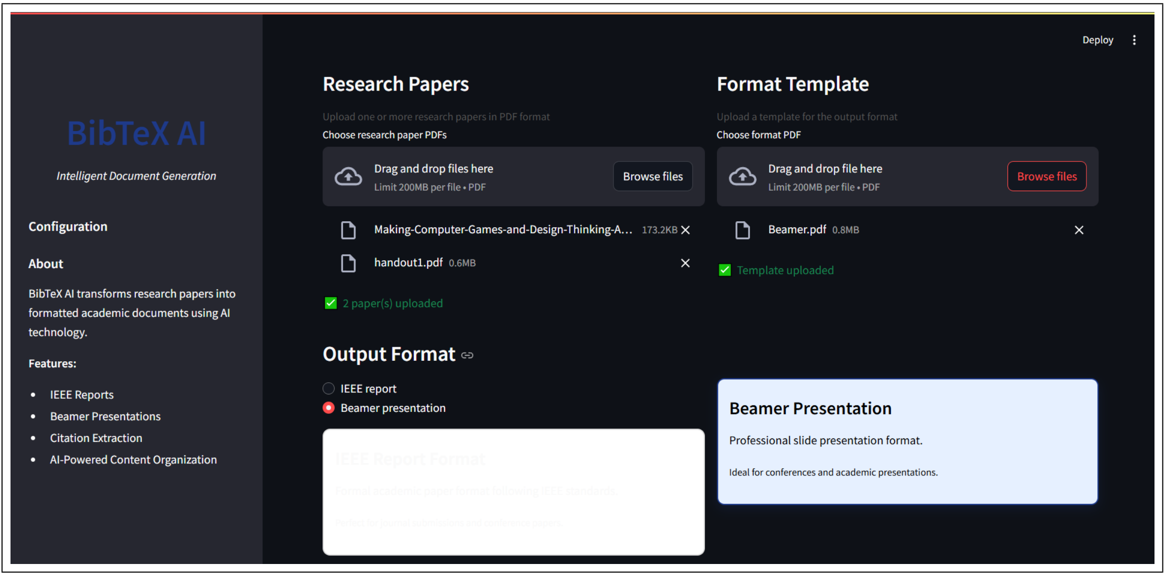
Fig. 7.1: BibTex AI Use Interface

The graphical interface, as shown in Figure 7.1, has been designed with a clean and intuitive layout. The user is provided with options to upload:

- **Research Papers (PDF)**
- **Desired Format PDF** (IEEE or Beamer)

Additionally, there are buttons to process the input and download the output in `.tex` format. The interface supports both input types through drag-and-drop or file browsing. This simplifies interaction, especially for users unfamiliar with LaTeX syntax or document structuring.

7.3 Input Format: Beamer Presentation



BibTeX AI

Intelligent Document Generation

Configuration

About

BibTeX AI transforms research papers into formatted academic documents using AI technology.

Features:

- IEEE Reports
- Beamer Presentations
- Citation Extraction
- AI-Powered Content Organization

Research Papers

Upload one or more research papers in PDF format

Choose research paper PDFs

Drag and drop files here
Limit 200MB per file • PDF

Making-Computer-Games-and-Design-Thinking-A... 173.2KB

handout1.pdf 0.6MB

✓ 2 paper(s) uploaded

Format Template

Upload a template for the output format

Choose format PDF

Drag and drop file here
Limit 200MB per file • PDF

Beamer.pdf 0.8MB

✓ Template uploaded

Output Format

○ IEEE report
● Beamer presentation

IEEE Report Format

Formal academic paper format following IEEE standards.

Perfect for journal submissions and conference papers.

Beamer Presentation

Professional slide presentation format.

Ideal for conferences and academic presentations.

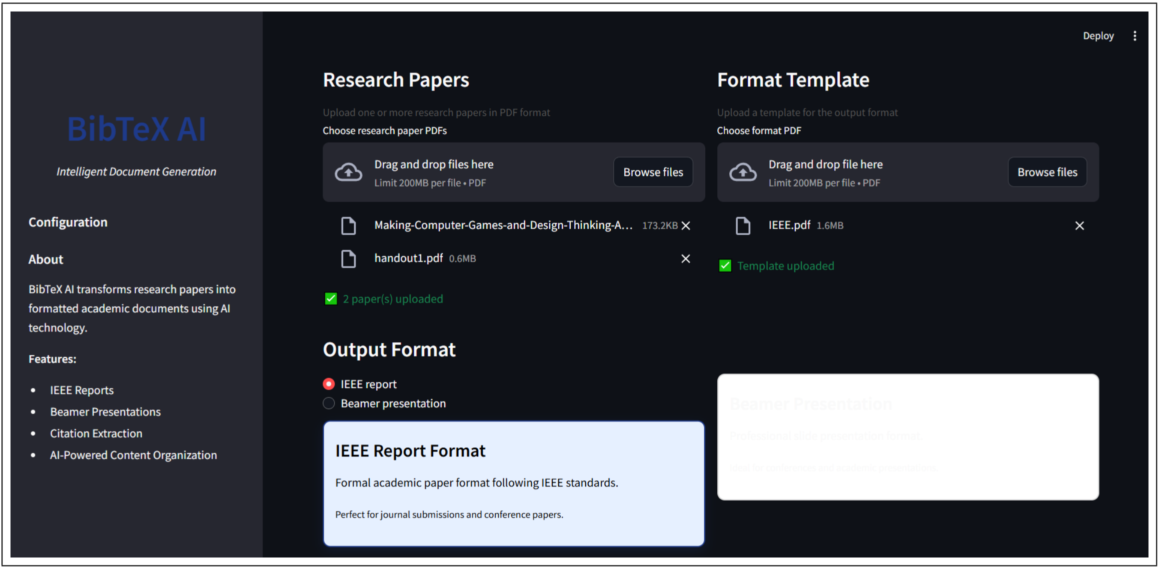
Fig. 7.2: Beamer Format Input

Figure 7.2 showcases an example of the Beamer presentation format uploaded by the user as a guide. The system extracts the structure of this format, such as:

- Title slide layout
- Section and subsection patterns
- Theme colors and font usage

This allows BibTeX AI to replicate the desired aesthetic and structural presentation when generating content based on research papers.

7.4 *Input Format: IEEE Paper*


Deploy

BibTeX AI

Intelligent Document Generation

Configuration

About

BibTeX AI transforms research papers into formatted academic documents using AI technology.

Features:

- IEEE Reports
- Beamer Presentations
- Citation Extraction
- AI-Powered Content Organization

Research Papers

Upload one or more research papers in PDF format

Choose research paper PDFs

Drag and drop files here
Limit 200MB per file • PDF
Browse files

Making-Computer-Games-and-Design-Thinking-A... 173.2KB

handout1.pdf 0.6MB

✓ 2 paper(s) uploaded

Format Template

Upload a template for the output format

Choose format PDF

Drag and drop file here
Limit 200MB per file • PDF
Browse files

IEEE.pdf 1.6MB

✓ Template uploaded

Output Format

● IEEE report
○ Beamer presentation

IEEE Report Format

Formal academic paper format following IEEE standards.

Perfect for journal submissions and conference papers.

Beamer Presentation

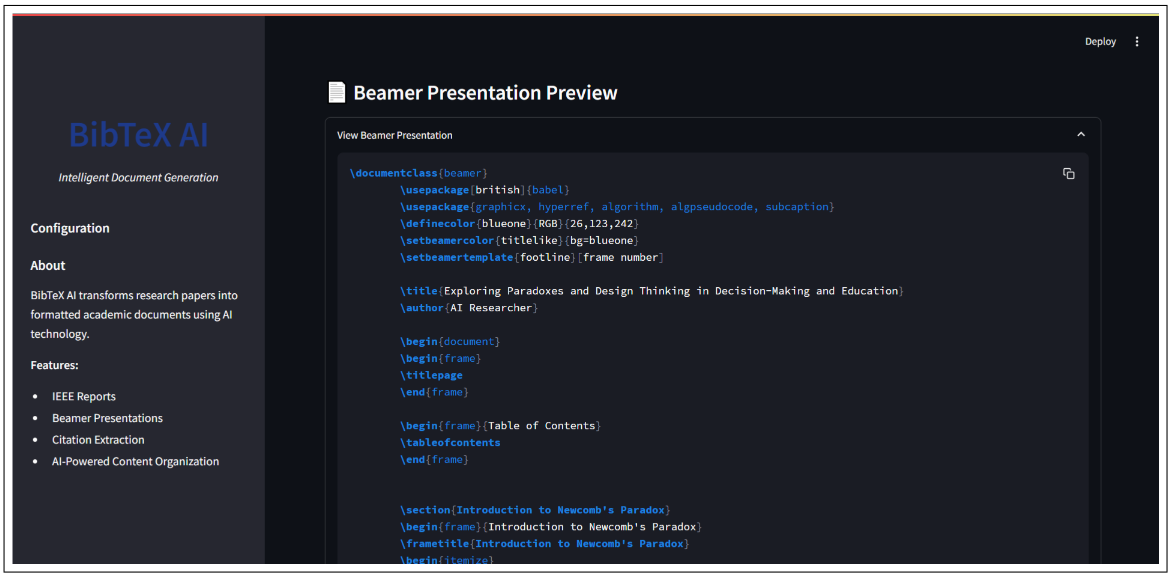
Professional slide presentation format.

Ideal for conferences and academic presentations.

Fig. 7.3: IEEE Format Input

As shown in Figure 7.3, the IEEE format input contains typical elements such as a two-column layout, abstract, keywords, section headers (e.g., Introduction, Methodology), and bibliography format. This input acts as a visual template that informs the system of the required LaTeX styling and layout, ensuring IEEE-compliant generation.

7.5 Output: Beamer Presentation



BibTeX AI transforms research papers into formatted academic documents using AI technology.

Features:

- IEEE Reports
- Beamer Presentations
- Citation Extraction
- AI-Powered Content Organization

Beamer Presentation Preview

```
\documentclass{beamer}
        \usepackage[british]{babel}
        \usepackage{graphicx, hyperref, algorithm, algpseudocode, subcaption}
        \definecolor{blueone}{RGB}{26,123,242}
        \setbeamercolor{titlelike}{bg=blueone}
        \setbeamertemplate{footline}[frame number]

        \title{Exploring Paradoxes and Design Thinking in Decision-Making and Education}
        \author{AI Researcher}

        \begin{document}
        \begin{frame}
        \titlepage
        \end{frame}

        \begin{frame}{Table of Contents}
        \tableofcontents
        \end{frame}


        \section{Introduction to Newcomb's Paradox}
        \begin{frame}{Introduction to Newcomb's Paradox}
        \frametitle{Introduction to Newcomb's Paradox}
        \begin{itemize}
```

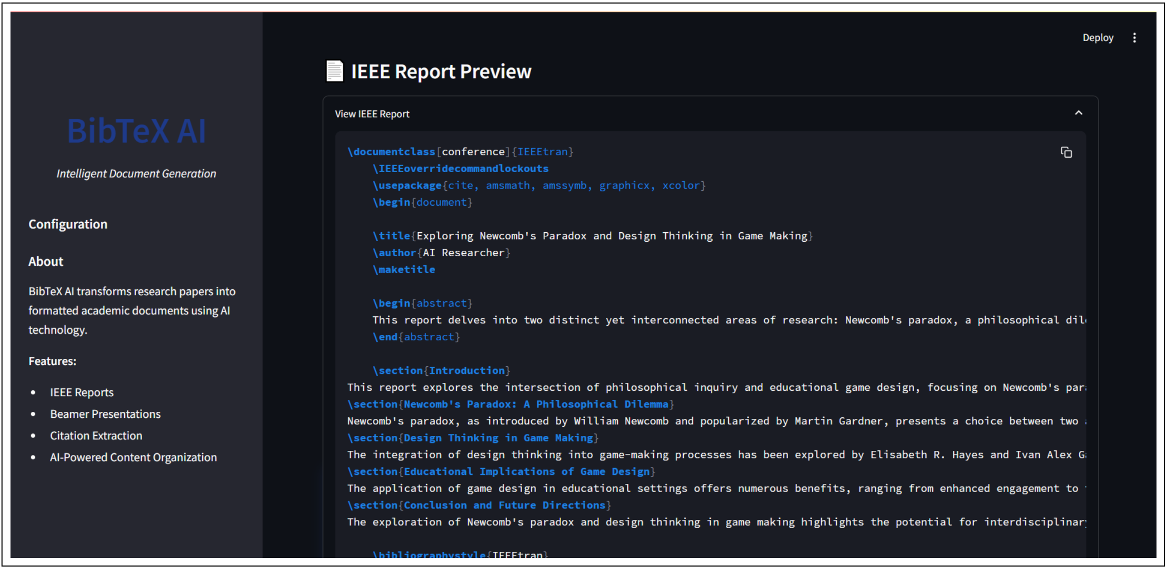
Fig. 7.4: Output Generated – Beamer Format

The output generated in the Beamer format, as seen in Figure 7.4, shows the content extracted from the research paper mapped into the predefined presentation structure. The generated `.tex` file:

- Automatically includes relevant sections (e.g., Motivation, Methodology, Results)
- Maintains consistent styling with the input Beamer format
- Integrates auto-formatted citations and BibTeX entries

The system demonstrates an ability to preserve both structural integrity and stylistic consistency, making the output ready for academic presentations.

7.6 *Output: IEEE Report*



BibTeX AI

Intelligent Document Generation

Configuration

About

BibTeX AI transforms research papers into formatted academic documents using AI technology.

Features:

- IEEE Reports
- Beamer Presentations
- Citation Extraction
- AI-Powered Content Organization

IEEE Report Preview

```
\documentclass[conference]{IEEEtran}
    \IEEEoverridecommandlockouts
    \usepackage{cite, amsmath, amssymb, graphicx, xcolor}
    \begin{document}

    \title{Exploring Newcomb's Paradox and Design Thinking in Game Making}
    \author{AI Researcher}
    \maketitle

    \begin{abstract}
    This report delves into two distinct yet interconnected areas of research: Newcomb's paradox, a philosophical dil
    \end{abstract}

    \section{Introduction}
This report explores the intersection of philosophical inquiry and educational game design, focusing on Newcomb's par
\section{Newcomb's Paradox: A Philosophical Dilemma}
Newcomb's paradox, as introduced by William Newcomb and popularized by Martin Gardner, presents a choice between two
\section{Design Thinking in Game Making}
The integration of design thinking into game-making processes has been explored by Elisabeth R. Hayes and Ivan Alex G
\section{Educational Implications of Game Design}
The application of game design in educational settings offers numerous benefits, ranging from enhanced engagement to
\section{Conclusion and Future Directions}
The exploration of Newcomb's paradox and design thinking in game making highlights the potential for interdisciplinar

    \bibliographystyle{IEEEtran}
```

Fig. 7.5: Output Generated – IEEE Format

In Figure 7.5, the LaTeX output generated by the system conforms to the IEEE standards. The document includes:

- Abstract and keyword section aligned with IEEE style
- Numbered sections (e.g., Introduction, Literature Review)
- Formatted citations and references using BibTeX

7.7 *Evaluation Metrics*

To assess the quality and effectiveness of the BibTeX AI system's output, we utilized **BERTScore**[10] to compare the system-generated content with reference research papers. BERTScore evaluates the semantic similarity between the generated text and reference texts based on contextual embeddings from pre-trained models[11][12]. The metrics calculated during the evaluation process are as follows:

7.7.1 *Precision*

Precision measures how many of the generated tokens match the reference tokens. It captures the relevance of the generated content relative to the reference text.

- **Average Precision**: 0.8023
  This score indicates that approximately 80.23% of the generated tokens are relevant and aligned with the reference content.

7.7.2 Recall

Recall measures how many of the reference tokens are captured by the generated content. It evaluates the system's ability to reproduce the information present in the reference text.

- **Average Recall**: 0.7877
  This score indicates that 78.77% of the content from the reference text was captured by the generated output.

7.7.3 F1 Score

The F1 score is the harmonic mean of Precision and Recall, providing a balanced metric that reflects both the relevance and completeness of the generated content.

- **Average F1 Score**: 0.7950
  This score suggests that the generated output provides a good balance between precision and recall, with a solid overall performance.

7.7.4 Evaluation Procedure

The evaluation was conducted using two research papers as references and the corresponding system-generated output[13][14]. The text was extracted from the PDF documents using `PyMuPDF` and compared using the BERTScore metric with `Roberta-large` as the pre-trained language model. Despite a warning that some weights were not initialized, the model was still able to generate meaningful evaluation scores[15].

CHAPTER 8
CONCLUSION

The BibTeX AI project showcases an important leap towards making academic document preparation easier with intelligent automation. Utilizing the power of large language models and a multi-agent system, the tool efficiently converts unstructured research papers into structured LaTeX content acceptable for both IEEE-style reports and Beamer presentations. This system not only simplifies the process of content creation but also solves typical problems of students and researchers in formatting and citation management. With its easy-to-use interface and visual input format support, BibTeX AI fills the gap between technical document production and user accessibility with different levels of LaTeX proficiency.

The system's capability to summarize key points, compare research papers, and cite using BibTeX indicates its real-world application in academic and professional environments. The modularity of the system ensures that integrating new agents or templates for formats is simple, making it easy to scale for future development. The output reflects the tool's ability to withstand varying formatting standards while upholding academic integrity.

During the process of development, several challenges were met and resolved through accurate PDF parsing, replicating formats, and validating LaTeX syntax. These experiences helped enhance insight into structuring content, managing errors, and optimizing user experiences. The product at completion reflects a judicious balance between the intelligence of AI and the classic LaTeX workflow.

In summary, BibTeX AI is a potential vehicle for automating the creation of academic content, minimizing the manual effort, and maximizing productivity levels for researchers. Future development could involve enhancing compatibility with more formats, providing features for collaborative editing, and supporting multiple languages, further expanding its influence in academic writing.

The BERTScore evaluation shows that the BibTeX AI system generates output that is semantically similar to the reference texts. With an F1 score of 0.7950, the system demonstrates a strong ability to reproduce accurate content, with room for further improvement, particularly by fine-tuning the underlying language models.

The result is a fully structured `.tex` document that can be directly compiled on platforms such as Overleaf or TeXStudio, significantly reducing manual formatting time for the user. The generated output validates the core functionality of **BibTeX AI**. By leveraging multi-agent collaboration, the system successfully transforms unstructured academic content into domain-compliant LaTeX documents. These results reinforce the system's potential in aiding students, researchers, and professionals in generating publication-ready content efficiently.

CHAPTER 9
FUTURE ENHANCEMENT

While the BibTeX AI tool successfully automates content production of academic material to LaTeX-based reports and presentations, there is enormous scope to further enhance its functionality. One such key area of enhancement is in refining the quality and level of generated content. Through the use of more sophisticated extraction methods, the system can extract more refined and sophisticated information from the research paper inputs. This would enable the final output—a report or Beamer presentation—to contain richer, more complete scholarly value.

The other key area of improvement relates to the inclusion of more potent large language models (LLMs) to address specific subtasks in the system. Using models like GPT-4 Turbo or Claude 3 can tremendously improve the coherence, organization, and technical soundness of the produced output. Additionally, by embracing a hybrid LLM approach—leveraging GPT-4 Turbo for formal structuring, Claude 3 for precise technical authoring, and Gemini 1.5 for comprehension of overall context—the system can strategically distribute tasks to models most appropriate for each task. This modular approach to specialized LLMs could potentially increase the overall quality and applicability of the produced scholarly documents.

Also, improving the system's capacity to process more research paper inputs can extend its use in research-heavy situations. Currently, the system has to limit the number of papers it can process because of computation limitations. Yet, with the addition of premium access to high-capacity LLMs, the system can be scaled to handle multiple papers simultaneously, process more pages per document, and provide comparative insights with deeper context and background. This would greatly serve users who want to synthesize vast amounts of research for academic writing or review articles.

In short, future BibTeX AI versions may advance toward a smarter, scalable, and scholarly more refined tool through enhancing content quality, judiciously utilizing multiple LLMs, and larger input capacity. These additions will not just enhance usability but also validate the system's utility in scholarly, professional, and research environments.

BIBLIOGRAPHY

[1] Mittelbach, Frank, Ulrike Fischer, and Chris Rowley. ”LATEX Tagged PDF Feasibility Evaluation.” LaTeX Project, September (2020).

[2] Ayush Thakur. The art of prompting: Unleashing the power of large language models, arXiv preprint March (2024).

[3] Rasnayaka, Sanka, et al. ”An empirical study on usage and perceptions of llms in a software engineering project.” Proceedings of the 1st International Workshop on Large Language Models for Code. 2024.

[4] Jia, Qinjin, et al. ”On Assessing the Faithfulness of LLM-generated Feedback on Student Assignments.” Proceedings of the 17th International Conference on Educational Data Mining. 2024.

[5] Zheng, Hao, et al. ”PPTAgent: Generating and Evaluating Presentations Beyond Text-to-Slides.” arXiv preprint arXiv:2501.03936 (2025).

[6] Qiao, Shuofei, et al. ”Benchmarking Agentic Workflow Generation.” arXiv preprint arXiv:2410.07869 (2024).

[7] Zhang, Jiayi, et al. ”Aflow: Automating agentic workflow generation.” arXiv preprint arXiv:2410.10762 (2024).

[8] Liu, Zijun, et al. ”A dynamic LLM-powered agent network for task-oriented agent collaboration.” First Conference on Language Modeling. 2024.

[9] Jansson, Peter. ”Scientific writing with LATEX.” (2016).

[10] Zhang, Tianyi, et al. ”Bertscore: Evaluating text generation with bert.” arXiv preprint arXiv:1904.09675 (2019).

[11] Saadany, Hadeel, and Constantin Orasan. ”BLEU, METEOR, BERTScore: Evaluation of metrics performance in assessing critical translation errors in sentiment-oriented text.” arXiv preprint arXiv:2109.14250 (2021).

[12] Shukla, Sanidhya Madhav, Chandni Magoo, and Puneet Garg. ”Comparing Fine Tuned-LMs for Detecting LLM-Generated Text.” 2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON). IEEE, 2024.

[13] Bevilacqua, Marialena, et al. ”When automated assessment meets automated content generation: Examining text quality in the era of gpts.” ACM Transactions on Information Systems 43.2 (2025): 1-36.

[14] Hu, Taojun, and Xiao-Hua Zhou. ”Unveiling llm evaluation focused on metrics: Challenges and solutions.” arXiv preprint arXiv:2404.09135 (2024).

[15] Chu, KuanChao, Yi-Pei Chen, and Hideki Nakayama. ”A better llm evaluator for text generation: The impact of prompt output sequencing and optimization.” arXiv preprint arXiv:2406.09972 (2024).

[16] Lehmann, René. ”Towards Interoperability of APIs-an LLM-based approach.” Proceedings of the 25th International Middleware Conference: Demos, Posters and Doctoral Symposium. 2024.