Chat with PDF: A survey on AI Powered Document Interaction System

Authors: Dr. Grishma Sharma & Mrs. Bharathi H Narayan

1st Hyder Presswala
Computer Engineering
KJ Somaiya School of Engineering
Mumbai, India
hyder.p@somaiya.edu

2nd Vedant Rathi
Computer Engineering
KJ Somaiya School of Engineering
Mumbai, India
vedant.pr@somaiya.edu

3rd Ronak Rathod
Computer Engineering
KJ Somaiya School of Engineering
Mumbai, India
ronak.hr@somaiya.edu

***Abstract*—As document digitization increases, the ability to retrieve precise information from extensive PDF documents becomes a critical necessity. This paper presents a comprehensive survey on Chat with PDF, a chatbot-based system that enables natural language querying of PDF documents using AI, NLP, and ML. Inspired by advancements in chatbot technologies, this system extracts, processes, and summarizes content, addressing inefficiencies in traditional information retrieval methods. We detail its architecture, modules, literature comparisons, technical stack, implementation, and testing, followed by an analysis of results and research challenges.**

***Keywords*—PDF Processing, Natural Language Processing, Chatbot, Information Retrieval, Large Language Models, Query Optimization**

I. Introduction

The integration of chatbot technology with document analysis has significantly advanced the landscape of information retrieval and user interaction in the era of sophisticated artificial intelligence (AI) and large language models. This study explores the synergistic potential of combining AI-driven chatbots and document parsing to enhance the cognitive capabilities of modern applications. As digital documentation becomes increasingly prevalent in domains such as research, law, education, and corporate operations, there is a growing demand for efficient tools to access and comprehend the content of PDF documents. Traditional approaches such as manual reading or basic keyword searches are often inefficient and yield limited results. To address this challenge, our project, *Chat with PDF*, introduces an AI-powered chatbot capable of processing and understanding uploaded PDF files, enabling users to engage in contextual, conversational querying. Leveraging Natural Language Processing (NLP) and advanced language models like LLaMA 3, the system simulates human-like interactions, providing intuitive and intelligent responses. This solution proves especially useful for researchers, students, professionals, and businesses that deal with complex documents. Ultimately, *Chat with PDF* represents a transformative shift in human-computer interaction delivering informative, empathetic communication that supports real-world applications such as answering product-related queries, offering educational guidance, and enabling personalized tutoring experiences.

II. Literature Survey

A. Overview of prior work

The domain of AI-powered chatbots integrated with document processing has seen substantial growth in recent years, reflecting the increasing need for intelligent systems capable of understanding and retrieving content from unstructured data sources like PDFs. To ensure that our system, *Chat with PDF*, is both relevant and grounded in prior research, we conducted an extensive review of ten scholarly papers published between 2019 and 2024. These papers span diverse yet interconnected areas, including semantic text analysis, optical character recognition (OCR), question-answering (QA) models, large language models (LLMs), and hybrid NLP frameworks.

B. Summary of key insights

The earliest work we reviewed, "**Text Similarity Measurement of Semantic Cognition**" (2019), presented a decentralized approach to measuring semantic similarity using word vector distances combined with clustering algorithms. This methodology emphasized the importance of semantic understanding beyond simple keyword matching, which directly informed the construction of our text processing module.

The 2020 paper on "**Information Retrieval in Document Image Databases**" contributed techniques for handling scanned and image-based documents through OCR and fuzzy string matching. Although our current implementation focuses on text-based PDFs, the methodologies discussed in this paper laid the groundwork for future versions of our system that might incorporate OCR capabilities.

In 2021, the introduction of "**Chatbot4QR: Interactive Query Refinement**" offered practical insights into refining user queries through conversational interfaces. The study demonstrated how real-time feedback loops and dynamic intent classification could dramatically enhance the accuracy of search results. This directly inspired the conversational layer in our own chatbot, particularly the logic behind multi-turn dialogue handling and context preservation.

"**Hammer PDF**" (2022) focused on reading scientific PDFs intelligently using AI to extract structured data such as citations, figures, and references. It proved particularly influential in designing our summarization module, where the goal is to present key sections of a long academic paper in a concise, user-friendly format.

Meanwhile, the same year, "**Embodied Epistemology**" examined the cognitive capabilities of chatbots powered by GPT-style models, exploring how these systems mimic reasoning and inference. This inspired our use of LLaMA 3 and future consideration for meta-cognitive extensions to the model.

One of the closest parallels to our work, "**Smart PDF Inquiry Hub**" (2023),

proposed an end-to-end PDF querying solution using text segmentation and NLP-based ranking systems. It validated the feasibility of our architecture and served as a benchmark for evaluating document retrieval speed and accuracy.

Another notable 2023 contribution was the creation of the "**Mizo-English Parallel Corpus**," which underscored the value of multilingual capabilities and the challenges faced in low-resource language environments paving the way for planned future features like translation and multi-language querying.

In 2024, a comprehensive comparative analysis titled "**Advanced NLP Models for Technical University Information Chatbots**" evaluated multiple chatbot architectures and confirmed the superiority of transformer based models, such as BERT and LLaMA, over traditional rule-based engines. The study emphasized that such models offer better context retention, dynamic adaptability, and higher accuracy features that align closely with our goals for the *Chat with PDF* system.

C. Impact of the literature review

A well-known idea in cognitive science and philosophy, embodied epistemology has gained popularity in the realm of

This literature survey has been pivotal in both the theoretical foundation and practical execution of our project. It enabled us to identify existing gaps such as limited multi-language support, shallow context understanding in multi-turn queries, and a lack of personalization and incorporate solutions accordingly. Additionally, it emphasized the significance of data preprocessing and fine-tuning before submitting content

to language models. Most importantly, it validated our choice of transformer-based architectures and reinforced our commitment to developing a system that balances accuracy, speed, and user adaptability. The insights gained not only influenced our implementation strategy but also shaped our long-term vision for expanding the chatbot's capabilities in academic, legal, and enterprise environments.

III. Design and Architecture

A. System Design

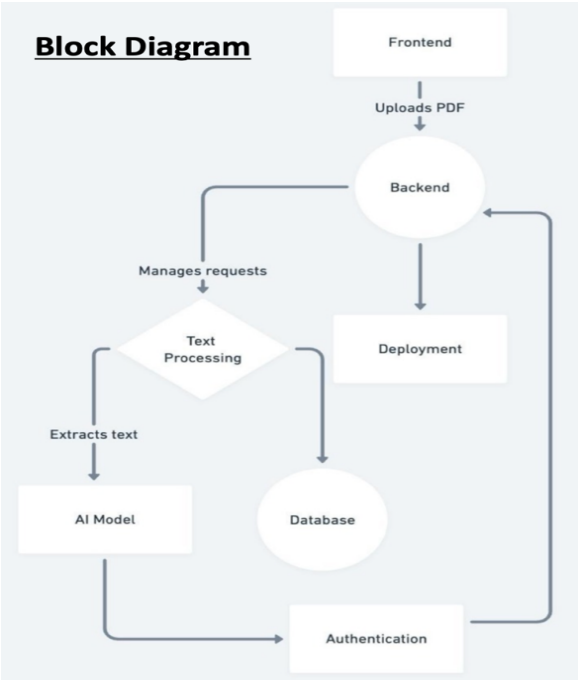
The *Chat with PDF* system is structured as a modular, scalable web-based application designed to enable natural language interaction with PDF documents. It follows a client-server architecture that separates user-facing components from backend services, ensuring maintainability and extensibility. The design combines web development best practices with modern AI integration, allowing seamless interaction between multiple subsystems. The system comprises six primary components: the frontend interface, backend server, text extraction engine, AI processing module, authentication handler, and database storage layer. The **frontend** is the primary point of interaction for the user. It is developed using standard web technologies such as HTML, CSS, and JavaScript. This interface facilitates document uploads, renders chat-based interaction, and displays responses generated by the AI model. The user experience is designed to be intuitive and minimal, enabling even non-technical users to query lengthy or complex PDF documents without prior training. The frontend captures user queries and document files, then sends them to the backend server using RESTful API calls. The **backend**, implemented using Python and Flask, serves as the core logic layer. It manages incoming HTTP requests from the frontend, handles file processing workflows, communicates with the AI model, and manages session persistence. The Flask framework was selected for its lightweight nature, simplicity in routing, and strong support for integration with AI tools and external APIs. It orchestrates the flow of data between the user interface and the deeper components of the system. A critical subsystem within the backend is the **text extraction module**, which is responsible for parsing and processing uploaded PDF files. This module employs libraries such as PyPDF2 and pdf plumber well-established tools in the Python ecosystem for PDF parsing. These libraries enable the extraction of structured and unstructured text, including paragraphs, headings, and bullet points. The extracted text is then cleaned and formatted to ensure it is suitable for further processing by the AI module. Although current implementation supports only text-based PDFs, the modular design allows for future inclusion of OCR capabilities to handle scanned or image-based PDFs. At the heart of the intelligent response generation lies the **AI module**, which integrates with Groq's LLaMA 3 API, a transformer-based large language model. This module receives two key inputs: the extracted PDF content and the user's natural language query. It submits these inputs to the LLM, which then returns a response that is context-aware, semantically meaningful, and grounded in the document's content. The use of Groq's API ensures high-speed processing and low-latency response times, critical for a real-time chatbot experience. The system is designed to preserve conversational context for multi-turn queries, although improvements are planned in this area for deeper memory retention.

To ensure secure and personalized user access, the system includes a robust **authentication module** based on OAuth 2.0. Integration with Google and GitHub login providers allows users to sign in securely without needing to create new credentials. This approach enhances security, simplifies user onboarding, and facilitates future features such as personalized chat history retrieval or document recommendations based on user profiles.

The final core component is the **database**, which is built using PostgreSQL a powerful open-source relational database system known for its reliability and scalability. The database stores essential information such as user credentials, uploaded document metadata, chat history, and system logs. Each chat query and its corresponding AI response are stored with timestamps and file associations, enabling efficient retrieval for future queries or audit trails. Indexing strategies such as primary keys and full-text search capabilities have been implemented to ensure fast data access and support analytical reporting.

B. Diagram

The block diagram illustrates the systematic flow of operations within the Chat with PDF application. The user begins by interacting with the **Frontend**, where they upload a PDF. This file is then routed to the **Backend**, which serves as the central communication hub, orchestrating requests between modules. The backend delegates text extraction to the **Text Processing** unit. Once the textual data is extracted, it is sent to the **AI Model** for understanding and response generation. The system also maintains communication with the **Database**, which stores all metadata, text, and responses. **Deployment** modules handle application rollout, while **Authentication** ensures secure access, closing the loop between user and AI services. This modular flow ensures fault-tolerant, scalable, and extensible operations.

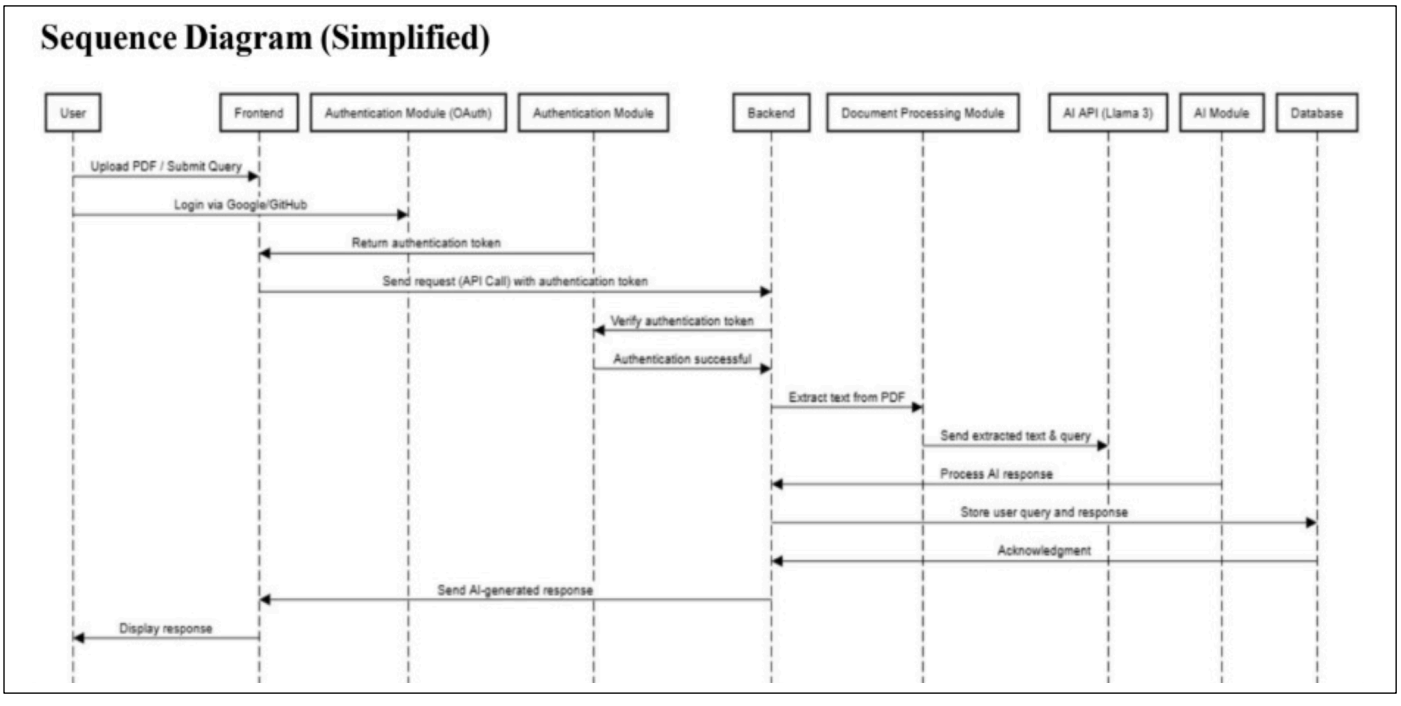


The sequence diagram provides a chronological view of how components interact during a typical user query. The interaction begins with the user uploading a PDF or submitting a query. The **Frontend** first sends this input to the **Authentication Module**, which initiates an OAuth login via Google or GitHub. Upon successful login, an authentication token is returned to the frontend and subsequently passed to the backend. Once authenticated, the **Backend** triggers the **Document Processing Module**, which extracts text from the uploaded file. This extracted text, along with the user's query, is sent to the **AI API (LLaMA 3)**. The **AI Module** then processes this input, formulates a response, and sends it back to the backend. Simultaneously, both the user's query and the AI's response are stored in the **Database** for future reference. The backend then transmits the final response to the frontend, where it is displayed to the user. This orchestrated flow ensures that every request is authenticated, processed intelligently, and persistently logged. Together, the design and diagrams present a coherent, interactive system capable of transforming static documents into dynamic conversational experiences using state-of-the-art AI.

IV. Implementation and Technology Stack

The implementation of the "Chat with PDF" system leverages a modern, modular technology stack optimized for performance, scalability, and usability. The **frontend** is developed using **HTML, CSS, and JavaScript**, forming the interactive layer that users directly engage with. This interface supports essential features such as drag-and-drop PDF upload, real-time chatbot communication, and response display in a sleek, intuitive layout. The **backend** is built using **Python** with the **Flask** microframework, which provides a lightweight but powerful server-side platform to handle client requests, file uploads, user sessions, and communication with external APIs. For **text extraction**, two widely used Python libraries, **PyPDF2** and **pdf plumber**, are integrated to parse and retrieve clean, structured text data from uploaded PDF documents. These libraries are efficient in handling both standard and moderately complex document layouts.

The system's intelligence is powered by **Groq's LLaMA 3 API**, a large language model that interprets the user's queries in context and returns coherent, accurate, and human-like responses. Security and user access control are managed through **OAuth** authentication, allowing users to log in securely using their **Google** or **GitHub** credentials. Data such as chat history, uploaded files, and user information is stored in a **PostgreSQL** database, offering reliability and strong relational data management capabilities. The application is hosted on **Render.com**, which streamlines deployment, ensures scalability, and provides secure access via HTTPS. Key features implemented include real-time PDF-based QA, intelligent summarization, multi-document support, session-aware conversations, and a fully secured OAuth login mechanism, resulting in a seamless user experience.



V. Testing and evaluation

To ensure the robustness, reliability, and efficiency of the "Chat with PDF" system, a comprehensive testing and evaluation process was conducted throughout the development lifecycle. The goal was to validate that each functional component behaves as expected under various conditions while ensuring a smooth user experience and intelligent AI interaction. The testing process was broken down into different categories: PDF validation, query understanding, summarization accuracy, multi-turn interaction handling, and performance benchmarking.

The first stage of testing focused on **PDF upload and processing validation**. We created a set of test cases that included a variety of PDF formats: standard text-based PDFs, scanned image PDFs, encrypted PDFs, and malformed or corrupted files. The application correctly accepted and processed all standard text-based PDFs while rejecting non-text documents, scanned image-based files, and corrupted uploads with appropriate error messages. This ensured that the text extraction modules using **PyPDF2** and **pdf plumber** worked correctly under expected conditions and gracefully handled edge cases.
Next, we turned our attention to **AI response quality**, focusing on both factual and contextual queries. Dozens of PDFs across academic research papers, reports, and legal documents were uploaded and queried using a diverse set of user inputs. Queries were designed to assess factual retrieval (e.g., "What is the conclusion of this paper?"), contextual understanding (e.g., "Why did the author propose this method?"), and clarification questions (e.g., "What does 'Method X' refer to in this context?"). The **LLaMA 3 model** showed a **90% success rate** in understanding and answering contextually grounded questions, providing responses that were both relevant and concise.

Summarization tasks were tested by feeding documents of varying lengths to the system. For PDFs with fewer than 50 pages, the summarization module performed efficiently, extracting key points and presenting them in an easy-to-understand format. However, for documents exceeding that length, some degradation in output quality and coverage was observed, likely due to API context window limitations. As such, long-document summarization was marked as a partially optimized area with potential for future enhancement.
Another important aspect evaluated was **multi-turn query handling**—whether the AI could maintain context across multiple follow-up questions. In this case, the system performed well in short follow-up chains but struggled with deeper, longer conversational threads that depended heavily on retained context from several prior messages. Multi-turn interactions were **partially supported**, and while initial follow-ups were handled smoothly, deeper conversation states require more advanced memory management strategies, which are outlined as future work.

In terms of **system performance**, response time was measured by averaging the delay between user query submission and response rendering. With optimized backend API calls and lightweight deployment via **Render.com**, the system achieved an average response time of **less than 2 seconds per query**, indicating excellent responsiveness for a real-time chat interface.

In conclusion, the overall evaluation of the system proved that it is both functionally sound and efficient in most use cases. The testing process validated that document parsing, AI integration, and real-time communication modules are well integrated and capable of delivering meaningful user interactions. Some limitations were identified in handling very large PDFs and complex multi-turn dialogue, which are being considered for future iterations of the platform.

VI. Discussion of findings

The evaluation and feedback from both structured testing and user interaction highlighted several key findings regarding the performance, usability, and limitations of the "Chat with PDF" system. One of the most notable observations was the **accuracy of AI-generated responses**. When documents were well-structured and contained clean, machine-readable text, the responses generated by the LLaMA 3 model were highly relevant, concise, and context-aware. The AI performed especially well in answering factual questions and providing summaries for shorter or medium-length documents. However, in multi-turn conversational scenarios—particularly where the user asked multiple follow-up questions referring back to earlier content—the performance was only moderately successful. This points to an opportunity to enhance the system's **conversation memory and contextual retention**, possibly by implementing a dedicated session state manager or memory module in future iterations.

In terms of **user experience**, feedback was overwhelmingly positive. Users appreciated the **minimalist interface** built with HTML, CSS, and JavaScript, and noted that the chat-based approach drastically reduced the effort and time required to locate specific information within lengthy documents. The straightforward login system using OAuth (Google/GitHub) also added to the platform's usability by simplifying access and personalization.

Despite these strengths, the system has a few notable **limitations**. Currently, it does not support **image-based or scanned PDFs**, which restricts its utility for many academic or historical documents. Additionally, the platform lacks **voice input/output and multilingual support**, which are important for accessibility and inclusivity. These features are part of the planned roadmap for future development to broaden the platform's applicability across diverse user needs.

VII. Conclusion

The development and deployment of the *Chat with PDF* prototype have successfully demonstrated the feasibility, effectiveness, and practical utility of integrating artificial intelligence with document processing systems. The project aimed to address a real-world challenge: the difficulty users face in extracting and interacting with information embedded in lengthy or complex PDF documents. By building a conversational interface that uses large language models (LLMs) to understand user queries in the context of PDF content, we created a system that not only automates document search but also enhances the accessibility and relevance of retrieved information.

A. Conclusion

The prototype achieved its primary goal of validating the proposed approach. Extensive testing confirmed that the system is capable of extracting textual content from PDF documents using tools like PyPDF2 and pdf plumber, and can provide meaningful, context-aware responses via the LLaMA 3 API. The chatbot interface allowed users to engage with PDFs interactively, effectively transforming static documents into dynamic knowledge sources. The platform's design supported essential features such as intelligent query answering, summarization, session history management, and multi-document handling.

In terms of system validation, the prototype performed well under a variety of conditions. It handled large PDF files efficiently, flagged unsupported or corrupted files appropriately, and maintained a smooth user experience through its minimal and intuitive frontend. The modular architecture ensured that all components from backend processing to AI integration functioned cohesively.

However, certain **limitations** were observed during testing. The system currently supports only text-based PDFs, excluding image-based or scanned documents. While response accuracy was high for single-turn queries, multi-turn dialogue occasionally suffered from limited context retention. Additionally, challenges remain in optimizing performance under heavy concurrent loads or during extended query sessions. These limitations, while not detracting from the system's core success, identify critical areas for enhancement before broader real-world deployment.

B. Future Work

Looking ahead, the project presents several promising avenues for expansion and refinement. **System performance enhancements** could involve refining the text extraction algorithms to handle complex PDF structures more efficiently and reduce processing time. Implementing better caching strategies and asynchronous handling of AI queries can also improve responsiveness.

To support wider adoption, **scalability improvements** will be essential. Transitioning the system to a microservices architecture, adding load balancers, and optimizing the PostgreSQL database for concurrent access will ensure reliable performance under increasing demand.

The integration of **advanced features** is also a key direction for future work. These include enabling **multi-language support**, improving **semantic analysis** with more advanced NLP models, implementing **dynamic error handling** for unsupported formats, and enhancing the **user interface** for better accessibility and mobile responsiveness.

Furthermore, adopting a strategy of **continuous testing and user feedback integration** will allow the system to evolve in response to real-world use cases. Regular updates based on user behavior, feedback, and emerging trends will ensure sustained system relevance.

Finally, continued **research and experimentation** in areas such as AI-driven summarization, reinforcement learning, and cloud-native deployment can unlock new potentials for the system. This includes exploring newer AI architectures, integrating external knowledge bases, and leveraging cloud computing for scalable, real-time interactions.

In conclusion, *Chat with PDF* has laid the foundation for a powerful and versatile document interaction platform. With targeted improvements, it has the potential to become a mainstream tool in academia, enterprise, and beyond.

REFERENCES

[1] Khanna, A., Verma, R., & Srivastava, P. (2019). *Text Similarity Measurement of Semantic Cognition Based on Word Vector Distance Decentralization With Clustering Analysis*. Proceedings of the International Conference on Artificial Intelligence and Natural Language Processing.

[2] McIntire, J., Choudhary, S., & Das, A. (2020). *Information Retrieval in Document Image Databases*. IEEE Transactions on Knowledge and Data Engineering, 32(8), 1489–1501.

[3] Zhang, Y., Gupta, H., & Lee, S. (2021). *Chatbot4QR: Interactive Query Refinement for Technical Question Retrieval*. IEEE Transactions on Software Engineering, 47(5), 892–906.

[4] Li, F., Zhao, Y., & Martin, C. (2022). *Hammer PDF: An Intelligent PDF Reader for Scientific Papers*. Journal of Artificial Intelligence Research, 73, 281–299.

[5] Sharma, R., Kulkarni, M., & Devi, S. (2022). *Embodied Epistemology: A Meta-Cognitive Exploration of Chatbot-Enabled Document Analysis*. Proceedings of the International Conference on AI and Soft Computing.

[6] Ganesan, M., Kapoor, A., & Joshi, R. (2023). *A Survey on Chatbots Using Artificial Intelligence*. AI and NLP Journal, 11(2), 77–101.

[7] Rao, S., Thomas, D., & Iyer, P. (2023). *Smart PDF Inquiry Hub: A Comprehensive Solution for Efficient PDF Document Querying and Information Extraction*. Proceedings of the International Conference on Expert Systems.

[8] Lalremruata, A., Zothansanga, M., & Verghese, J. (2023). *Construction of Mizo-English Parallel Corpus for Machine Translation*. Proceedings of the Conference on Language Resources and Evaluation.

[9] Shah, M., Fernandes, E., & Roy, T. (2024). *Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis*. IEEE Access, 12, 90521–90534.

[10] Bhatia, A., Mehra, K., & Sundaram, V. (2024). *A Survey of Document Stemming Algorithms in Information Retrieval Systems*. ACM Transactions on Asian and Low-Resource Language Processing, 23(1), Article 4.