**Mini Project Report on**



**Faculty BioGen**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Naman Gupta**  **2021319**

***Under the Mentorship of***

**Dr. Surender Singh Samant**

**Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January-2025**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Faculty BioGen”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Surender Singh Samant, Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name: Naman Gupta University Roll No.: 2021319

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-3** |
| Chapter 2 | Literature Survey | **4-6** |
| Chapter 3 | Methodology | **7-9** |
| Chapter 4 | Result and Discussion | **10-13** |
| Chapter 5 | Conclusion and Future Work | **14** |
|  | References | **15** |

**Chapter 1**

**Introduction**

**The Faculty Bio-Generation System is a new service designed to create and retrieve faculty profiles using advanced technology Faculty profiles are important to academic institutions, and provide insights into faculty members’ skills, achievements, and contributions. However, maintaining this information manually can be time-consuming and prone to inconsistencies. This project addresses these challenges by utilizing state-of-the-art tools and systems to painstakingly produce accurate, concise, and professional faculty bios.**

**The core of this project is a structured data set stored in a text file containing information about all faculty members, including their names, titles, research fields, achievements, and notable publications This data is generated using Hugging Face transformers are processed and converted into high-dimensional embeddings. These embedded features serve as interpretations of textual content, capturing relationships and contextual understanding, which are essential for successful retrieval and matching.**

**The vectorized data is indexed using FAISS (Facebook AI Similarity Search), a powerful vector database optimized for similarity searching and clustering. This allows the system to quickly retrieve relevant faculty data based on user queries. FAISS is efficient, enabling the application to handle large amounts of data while maintaining a fast response time.**

**For meaningful and user-friendly information, the system integrates the Large Language Model (LLM) via LangChain. LLM acts as the brains of the application, interpreting user queries, retrieving relevant data from vector databases, providing faculty BIOS with a well-configured LangChain system ensures smooth communication with LLM and between vector databases, enabling smooth flow of information and results.**

**Streamlit is used as a frontend framework to provide communication and friendly interaction. The web application allows users to enter questions about faculty members and access instant, professionally generated bios. Streamlit’s simplicity and flexibility makes it ideal for creating intuitive interfaces, ensuring that users can interact with the system effortlessly.**

**This application represents a unique blend of modern technologies, including natural language processors, vector databases, and web-based interfaces. The combination of Hugging Face Transformer, FAISS, LangChain, and Streamlit ensures that the system is not only highly efficient but also highly scalable and adaptable to educational settings. By automating the faculty biogenesis process, the Faculty Biogenesis System saves time, reduces errors, and increases the availability of faculty information, making it a valuable tool for universities and institutions of learning.**

**Chapter 2**

**Literature Survey**

**The Faculty Bio-Generation System builds upon foundational advancements in natural language processing (NLP), vector databases, and user-centric web application frameworks. This section explores the key technologies and methodologies that have inspired and shaped this project, highlighting their significance in automating information retrieval and generation tasks.**

**2.1 Natural Language Processing and Large Language Models (LLMs):**

**Natural language processing has evolved rapidly with the advent of large-scale language models such as GPT and BERT, which are changing the way machines understand and process human-like text, and research by Vaswani et al. (2017) introduced the transformer architecture as a key component in NLP which enabled the model to capture remote dependencies and contextual relationships in text. Hugging Face, a widely recognized library for NLP, builds on this foundation, providing pre-trained transformer diagrams that are optimized for different tasks. This enhancement makes it possible to generate meaningful and coherent informational responses based on structured and unstructured data, and forms the backbone of the bio generative potential of this work.**

**2.2 Vector Databases for Efficient Data Retrieval:**

**The increasing complexity and volume of data has necessitated the development of efficient data retrieval systems. Traditional keyword-based search methods fail to capture semantic relationships between queries and data sets. FAISS (Facebook AI Similarity Search) emerged as a powerful solution for vector-based retrieval, enabling efficient similarity searches in large datasets Research demonstrated the effectiveness of FAISS in applications requiring high-quality data such as recommendation systems and semantic search is By using FAISS, this project ensures faculty information based vectorized embedding so is fast and accurate, which meets today’s requirements for speed and scalability.**

**2.3 Embedding Techniques and Semantic Understanding:**

**Embedding techniques play an important role in distinguishing between raw textual data and machine-readable data. Research on textual encoding techniques, especially sentence-book encoding, has shown that textual content can be represented as a complex vector. Hugging Face transformers provide state-of-the-art embedding capabilities, enabling the capture of subtle meaning and relationships in text. These embedding faculties form the basis of the bio-generation process, enabling the application to store and retrieve semantic data with great accuracy.**

**2.4 Integration Frameworks like LangChain:**

**LangChain, a framework designed to integrate LLMs with external data sources has gained attention for its ability to handle complex multi-dimensional business processes Recent research and applications have established LangChain's effectiveness in simple communication highlighting the gap between LLMs and vector databases. Its modular architecture allows for customization and scalability, making it an important tool in applications such as faculty bio-generation systems, where data retrieval and text generation must work in harmony.**

**2.5 Interactive Web Applications and Streamlit:**

**Interactivity is an integral part of any modern application, and frameworks such as Streamlit have made great strides in facilitating interactive web interfaces The research on user experience (UX) design emphasizes the importance of accessibility and responsiveness in web applications, especially academic and business -Streamlit's small architecture for devices and Support for rapid prototyping and these principles meet, and empower developers to create user-friendly interfaces This project uses Streamlit to present user-friendly interfaces, ensuring easy access to faculty by students, researchers and staff bios.**

**2.6 Applications in Academia:**

**The concept of academic performance automation has been extensively explored in recent literature. Studies have highlighted the potential of AI-powered tools to increase productivity in academic institutions, from automated grading systems to research reviews Faculty bio-generation programs contribute to work this growing presence is achieved by addressing a specific and influential need: building and maintaining faculty profiles. By integrating state-of-the-art technology, the system aligns with the academy’s goals of improving efficiency and reducing manual effort.**

**Chapter 3**

**Methodology**

**The Faculty Bio-Generation System was developed using a structured and modular approach, integrating advanced technologies to ensure seamless functionality and efficient performance. The methodology involved multiple stages, including data collection and pre-processing, vectorization, database management, query processing, and interface development. Each step was carefully designed and implemented to ensure accuracy, scalability, and user-friendliness.**

**3.1 Data collection and pre-processing**

**The first step in the development process was the development of detailed data on faculty descriptions. A structured text file was created to store information such as names, designations, departments, research areas, accomplishments, and notable publications. This data formed the basis of the entire system. The material was handwritten and arranged to be consistent and accurate.**

**Pre-processing methods were then used to clean and normalize the data. This includes eliminating redundant information, addressing missing values, and ensuring consistency. Pre-processing ensured that the input data was suitable for downstream operations such as vectorization and semantic searching.**

**3.2 Vectorization of faculty data**

**The next step was to convert the transcripts into a machine-readable format using a variety of embedding techniques. Hugging face transforms were used to create higher embeddings for each faculty entry. These embeddings capture the semantic relationships of the text, enabling the system to perform efficient and contextual retrieval.**

**Each faculty report was converted into a dense vector representation that included its semantic meaning. These vectors were stored in a vector database, forming the core of the system restoration. This step was crucial to ensure that the system could correctly understand and answer user questions.**

**3.3 Vector database**

**The vectorized data were stored and indexed using FAISS (Facebook AI Similarity Search), a state-of-the-art vector database optimized for similarity search. FAISS was chosen because of its ability to handle high-dimensional data efficiently, even in CPU-based systems. The database was structured to facilitate rapid and accurate retrieval of relevant faculty records based on user-friendly queries.**

**A vector database was designed to accommodate future changes to the dataset to ensure scalability. This modular design allows the program to grow and change as new faculty members join the institution or as existing records are updated.**

**3.4 Query Functions with LLMs and LangChain**

**The primary functionality of the system is based on its ability to handle user questionnaires and generate short faculty bios. This was achieved through LangChain with the addition of the Large Language Model (LLM). The LLM interprets user input, retrieves relevant vectors from the FAISS database, and produces human-like responses.**

**LangChain facilitated seamless communication between the LLM and the vector database, ensuring that data acquisition and response generation happened in a synchronized manner by quickly developing technology to guide the LLM to create a BIOS that is precisely and professionally. This presentation was designed to extract relevant information from the retrieved vectors and to condense them into a user-friendly form.**

**3.5 Development of web interfaces**

**To provide an intuitive and intuitive user experience, a web interface developed using Streamlit was incorporated into the system. Streamlit was chosen for its simplicity and flexibility, enabling rapid prototyping and deployment of interactive applications.**

**The interface allows users to enter questions about faculty members and view dynamically generated bios. Features such as search filters and session memory were added to improve usability. The design focused on ensuring that users can interact with the system effortlessly, regardless of their technical skills.**

**3.6 Deployment and Scalability Considerations**

**The system was deployed on a server environment capable of handling multiple user requests simultaneously. Streamlit’s deployment capabilities and the lightweight nature of FAISS ensured that the application could run efficiently without requiring extensive computational resources. Provisions were made to update the dataset and vector database periodically, ensuring that the system remains relevant and up-to-date.**

**Chapter 4**

**Result and Discussion**

**The Faculty Bio-Generation System successfully automates the generation and retrieval of faculty bios, demonstrating its efficiency, accuracy, and usability. The results obtained from testing the system across various scenarios highlight its ability to handle complex queries and provide concise, contextually relevant responses. This section discusses the system’s performance, key outcomes, and insights gained during its development and testing phases.**

**4.1 Result**

**4.1.1 Question accuracy and relevance:**

**The system proved to be highly accurate in retrieving faculty information and creating bios. Leveraging the hugging face transformer for embedding generation and leveraging FAISS for vector-based searches, the application consistently provided contextually appropriate answers to questions such as “Who is the head of the computer science department?”. or “While focusing on research contributions, Drs. Give John Doe again.” This resulted in more accurate and structured presentations.**

**4.1.2 Characteristics of generation processes:**

**The addition of Large Language Modeling (LLM) through Langchain enabled the system to generate human-like responses that were professional and easy to understand the rapid pace of technical infrastructure played a key role in guiding the LLM for relevant information was extracted and presented well. The development of the bios provided a balance between brevity and breadth, ensuring that it was appropriate for academic and professional purposes.**

**4.1.3 System quality and scalability:**

**The use of FAISS for vector indexing enabled the system to handle large data sets with minimal latency. Even with a dataset containing hundreds of faculty records, the average response time was less than two seconds, demonstrating the efficiency of the system Furthermore, the modular design ensures flexibility, and it allows the application to retrieve new faculty records or adapt to changes in data formats without significant reformatting.**

**4.2 Discussion**

**4.2.1 Strengths of the system:**

**Faculty bio-generation excels at automating a traditional manual and error-tying process. By combining advanced NLP techniques and an efficient retrieval strategy, the application provides accurate results and identifies relevant information. The modular architecture ensures flexibility, allowing for the easy integration of new features or data sources in the future. Using Streamlit for the front end simplifies deployment and ensures a responsive user interface.**

**4.2.2 Challenges:**

**Many challenges were encountered during development. One of the key issues was to ensure that the answers provided by the LLM were focused and appropriate, especially for ambiguous questions. This was addressed through iterative prompt engineering and extensive testing. Another challenge is checking for data inconsistencies in the original data. Preprocessing techniques were used to standardize the data and ensure consistency with the vectorization process.**

**4.2.3 Explore and learn:**

**The event highlighted the importance of integrating different technologies to achieve integrated solutions. The inclusion of Hugging Face Transformer, FAISS, LangChain, and Streamlit demonstrated the value of taking advantage of specialized tools for specific projects. An iterative development process that includes user feedback and rigorous testing is essential in refining the system and fixing potential weaknesses.**

**4.2.4 Possible improvements:**

**Although the system works well, there is still room for improvement. For example, adding multilingual support could expand usability, especially in academic settings. Adding a feature to customize a BIOS based on specific attributes, such as learning experience or recent publications, can further enhance its usefulness. Additionally, using role-based services can ensure data security and prevent unauthorized access to sensitive faculty information.**

**4.2.5 Real World Applications:**

**The program has tremendous academic and professional potential. University websites can use them to dynamically generate faculty profiles, ensuring that information is kept up to date. Academic conferences can use the system to quickly generate speaker profiles, while administrators can use it to streamline internal business processes. The ability to handle complex questions and provide detailed, professional bios makes the app a valuable tool for organizations looking to increase their productivity.**

**Chapter 5**

**Conclusion and Future Work**

**5.1 Conclusion**

**The faculty bio-generation program represents a significant step forward in automating the generation of academic data. Combining the power of LLM, vector databases, and agile engineering, the framework provides a highly responsive, contextual approach that increases academic productivity.**

**5.2 Future work:**

**5.2.1 Enhanced data integration:**

**Integrating APIs to automatically update faculty data from university records or research databases.**

**5.2.2 Multilingual support:**

**Extends the system to support multiple languages, allowing it to be used worldwide.**

**5.2.3 Improved UI/UX:**

**Improved web-based interface for ease of access and interaction.**

**5.2.4 Scalability:**

**Optimizing the system to handle large amounts of data with limited resources.**

**5.2.5 Advanced Features:**

**To include functionality such as sensitivity analysis, publication standards, or personalized recommendations based on faculty expertise.**

**References**

[1] <https://github.com/Namangupta123/Graphic_Chat_Bot>

[2] <https://graphicchatbot.streamlit.app/>

[3] <https://github.com/Vinayak-Sharma12/ThaparGPT>

[4] <https://www.langchain.com/langsmith>

[5]<https://huggingface.co/>

[6] <https://chatgpt.com/>

[7] **Johnson, J., Douze, M., & Jégou, H. (2017).** “Billion-scale similarity search with GPUs”

[8] <https://github.com/facebookresearch/faiss>

[9] <https://cohere.com/>