UNIVERSITY OF PADOVA

COMPUTER ENGINEERING *(ARTIFICIAL INTELLIGENCE AND ROBOTICS)*

DESIGN AND DEVELOPMENT OF A SECURE NLP-POWERED CHATBOT FOR CONFIDENTIAL DOCUMENT RETRIEVAL

SUPERVISOR

PROF. ANTONIO RODA

UNIVERSITY OF PADOVA

CANDIDATE

VISHAL KUMAR

DEPARTMENT

DEPARTMENT OF INFORMATION ENGINEERING

STUDENT ID

2048663

ACADEMIC YEAR

2024-2025

Abstract

This period is defined by processing that is data-driven. Systematic and secure retrieval of data from the confidential documents has become a crucial need for many organizations. These documents are stored mostly in PDF format. These contain important information that is required for various tasks like strategic planning, policymaking, regulatory auditing, and academic research. However, traditional methods of data retrieval from such documents are manual, which means they are more time-consuming and error-prone. The objective of this thesis is to present the design and development of a secure NLP-powered chatbot that can be used to extract and retrieve information from confidential PDF documents while also ensuring data privacy and contextual relevance.

The key innovation in this project is the use of modern NLP techniques that can be used to create a secure local deployment model. Firstly, the documents are analyzed by using a tool called pdfplumber. This tool extracts the text from PDF files. After that, the text is tokenized, which means it is broken down into smaller pieces or "chunks" using the Natural Language Toolkit (NLTK). Each chunk that is obtained is then turned into a numerical form called an embedding. It is done by using the Sentence Transformers library. These embeddings are essential for the system, as it is used to understand the meaning of the text.

These embeddings are stored using FAISS in a special index. This tool allows the user to search fast for the answers. When a user types a question, the system converts the question into an embedding. Then it compares it with the stored data to find the most relevant text for the response. Lastly, Mistral-7B is used, which is a powerful AI model. It generates a human-like response using the relevant text. The confidential data is ensured to never reveal the data to external servers, which maintains privacy and compliance with data protection policies.

This model is easily expandable and adaptable to various use cases. The individual components are the embedding model, search engine or response generator. These can be independently updated or replaced as per the need. This makes the model flexible. The model can be used in various domains like healthcare, finance and education. In these sectors, confidentiality must be handled with care.

The project shows the efficiency of the integration of advanced NLP models and secure computing techniques. It transforms static and inaccessible PDFs into interactive and

accessible knowledge systems. The message to future researchers is to work on more advanced privacy-preserving methods like differential privacy or federated learning. They can also expand the system to handle multimedia content like images, charts, or tables. These modifications would further enhance its applicability and usefulness in real-world scenarios. As the advancement in AI and data privacy regulations increases, the system is designed to stay adaptable and compliant, ensuring long-term sustainability and trust.

Contents

Listing of figures

Listing of tables

Listing of acronyms

Natural Language Processing ……..**NLP**

Artificial intelligence……**AI**

Portable Document Format……...**PDF**

Facebook AI Similarity Search…….**FAISS**

Natural Language Toolkit…….**NLTK**

Command Line Interface……..**CLI**

Graphical User Interface……..**GUI**

1

Introduction

1.1 BACKGROUND AND CONTEXT

The digital revolution has been observed in various sectors, which has resulted in the global usage of e-documents for storing, sharing, and securing crucial data. Portable Document Format (PDF) is favored for maintaining the integrity and format of text across platforms. Institutions in various sectors, including healthcare, academia, finance, law, and government, preserve repositories of PDFs containing critical data such as legal contracts, patient records, policy frameworks, and research data. The ability to retrieve specific data, which is lost within these repositories, is essential to review in a timely manner to ensure compliance, decision-making, and enhanced productivity.

PDF documents are used universally but show extensive challenges to retrieve information. The reason is that they are generally lengthy, unstructured, and hard to find without accurate keywords. As a result, traditional search methods, which require manual searching within PDF readers, are not efficient, as the user does not know the right words to use for the research. Also, in cases where documents are confidential, relying on cloud-based search can result in a serious risk of data privacy. There is a high demand for modern systems that are capable of analyzing large amounts of information as well as understanding user queries in natural language to deliver accurate and context-aware answers.

Natural Language Processing (NLP) gives a promising path for changing the interaction with textual data. By allowing the system to understand human language, NLP ensures the development of intellectual agents that can interpret the meaning of a user's query, find relevant data, and generate natural language responses. The NLP, together with a secure local deployment system, can provide smart search capabilities and full control over data privacy.

Recent advancements in transformer-based language models and neural semantic search techniques have provided a new wave of use in the data retrieval domain. Retrieval, which uses embedding, allows a more subtle understanding of both documents and queries. The model integrates modern NLP techniques with secure architectures to ensure that data privacy remains uncompromised throughout the document retrieval process.

NLP has proved to be a revolutionary technology in the era of artificial intelligence, which gives machines the ability to analyze, understand, and create human language in a way that is highly understandable, natural, and effective. The capabilities of NLP have been a basic implementer for a broad range of applications across various domains. The examples are the Internet of Things (IoT), document analysis, and broader AI-powered systems. The continuous improvements of NLP techniques are helping existing technologies to enhance and provide new frontiers for human-computer interaction and intelligent automation.

One of the most important applications of NLP is the designing of chatbots, which is coupled with IoT ecosystems. The increase in the use of IoT devices among smart households, healthcare, the industrial sector, and urban livelihoods has created a high demand for such intuitive interfaces that allow users to control and interact with these devices naturally. NLP-powered chatbots fulfill this high demand by converting human language into instructions for the devices in computer language. The traditional command-based systems require fixed input formats, and these translatory systems provide advanced NLP models to understand the intent of the user, manage dialogue context, and handle difficult or complicated requests. This results in easier and more human-like interactions, improving the usability and accessibility of IoT systems [1].

Beyond IoT, NLP has major significance in the sector of document analysis. It is a field related to the extraction and evaluation of information from unstructured text documents. The existing method of document processing generally has human review and data entry, which is difficult manual work and susceptible to human error. NLP increases the ability of the system to execute the process by applying suitable techniques like text classification, named entity recognition (NER), relationship extraction, and summarization. These techniques help various systems to identify key concepts, categorize documents by suitable classifications, and generate summaries to convey the essential information without requiring users to read lengthy texts [2].

There is a huge impact of NLP on document analysis in sectors such as healthcare, legal, finance, and academia, as the documents are in large quantity and the stakes for accuracy are high. Taking an example from the healthcare sector, clinical documents such as discharge summaries, radiology reports, and medical research articles contain important information that can influence a patient's treatment and results. NLP systems can provide diagnoses, medication information, and treatment plans automatically, which facilitates better record keeping and decision support. Similarly, in the legal formalities, NLP helps in contract review and legal

research by identifying relevant points and precedent cases, organizing workflows, and reducing time consumption [2].

There are huge advancements in NLP for document analysis, and these are accompanied by the integration of NLP within AI-powered systems. The enhancements have been made in machine learning, like deep learning and transformer-based systems like BERT and GPT. It has extensively improved NLP's ability, which is used to identify semantic meaning, context, and languages involved. These models learn from the vast text available, which enables generalization across different domains and languages. This means that AI systems can now perform complex linguistic tasks. The tasks include sentiment analysis, question answering, and language translation with unprecedented accuracy [3].

In addition to this, continuous improvements in unsupervised and traditional learning techniques are taking place. This will make NLP models more adaptable to specialized domains with limited training data, expanding their applicability.

NLP has become an essential technology in healthcare and clinical informatics, and so it is significantly transforming the way unstructured textual data are processed and utilized. As it is known that healthcare systems generate massive volumes of data on a daily basis, which includes clinical notes, radiology reports, case reports, and patient interactions. The main challenge is to extract valuable data from these texts. The reason is their unstructured nature, domain-specific terminologies, and variability in language use. However, recent advances in NLP have enabled progress in automating the analysis and retrieval of clinical information. This includes deep learning techniques, which enhance patient care and support decision-making.

One of the prominent applications of NLP is the automated detection of medical adverse events from the available text data. It is a critical task to ensure patient safety and quality assurance. Consider an example like this: the detection of complications such as total hip replacement dislocations are often buried within lengthy postoperative notes and have traditionally relied on manual chart reviews that are time-consuming and prone to human error. NLP systems can now identify these adverse events with remarkable accuracy. This can be done by employing deep learning models such as bidirectional Long Short-Term Memory (Bi-LSTM) networks combined with contextual word embeddings. This kind of model captures the subtle linguistic cues and contextual dependencies that signify a dislocation event, which outperforms previous

keyword- or rule-based methods [4]. The ability of NLP enables real-time surveillance and timely interventions that reduce morbidity associated with postoperative complications.

In another case, NLP applications designed to extract critical diagnostic information from provided reports have greatly benefited the field of radiology. Radiology reports often contain detailed observations that are essential for patient management, but they are locked in free-text form. NLP models are capable of detecting findings such as pulmonary embolisms or intracranial hemorrhage with high sensitivity, supporting faster and more accurate diagnoses [5]. These NLP techniques integrate syntactic parsing with domain-specific vocabularies and ontologies. This enables the transformation of narrative descriptions into structured data that can seamlessly integrate with electronic health records (EHRs). The structured data allows clinical decision support, research, and administrative tasks like coding and billing.

The recognition and normalization of domain-specific terminology is the fundamental challenge in clinical NLP. It is vital for ensuring consistent interpretation across diverse texts. Mining methods to recognize terms automatically reveal that a combination of linguistic rules and statistical metrics (such as term frequency-inverse document frequency (TF-IDF)) significantly improves the identification of relevant medical terms [6]. The advancements in terminology extraction support clinical NLP. It also contributes to building comprehensive medical ontologies.

Clinical case reports offer details of individual patient experiences and present a valuable yet underutilized resource. NLP approaches are tailored for mining these reports and employ named entity recognition (NER), relation extraction, and semantic linking. This converts narrative content into structured knowledge. This process enables the aggregation of clinical insights in various cases. It facilitates systematic reviews, and it aids evidence-based medicine [7]. The structured representation of case reports allows researchers and clinicians to identify patterns and rare adverse events that otherwise might be overlooked.

A new era of data-driven healthcare has been initiated, and it is done by the integration of NLP with other emerging technologies such as the Internet of Things (IoT). Health monitoring systems can provide a comprehensive and timely picture of patient status, which is done by combining sensor data streams with unstructured clinical notes processed via NLP. For instance, early detection of clinical deterioration in chronic disease patients is enhanced by analyzing physician notes and patient-reported symptoms alongside physiological data, which

enables personalized interventions and resource optimization [8]. The given fusion underscores the evolving role of NLP as a critical intermediary in health informatics ecosystems.

NLP-powered conversational AI platforms, or chatbots, have gained significant traction. These chatbots leverage a variety of NLP techniques, including rule-based dialogue management, retrieval-based responses, and generative models grounded in transformer architectures, to facilitate natural, context-aware interactions [9]. Chatbots assist users in the healthcare sector by answering health-related inquiries, scheduling appointments, and providing medication reminders. It reduces the burden on clinical staff and improves patient satisfaction. Advanced chatbots are also being designed to handle complex, multi-turn dialogues and exhibit empathy. This supports mental health interventions and chronic disease management. The continuous evolution of NLP has propelled these conversational agents toward greater usability and effectiveness, which is from simple pattern matching to deep learning-driven contextual understanding [10].

Document analysis, when powered by NLP, helps to automate the classification, entity extraction, and summarization of clinical texts. Automated classification helps organize documents by disease category or treatment type. This facilitates the rapid retrieval and secondary data uses like clinical research and billing. The quality and utility of EHRs has been enhanced by extracting key clinical entities such as medication names, dosages, and laboratory results. Summarizing the algorithms condenses lengthy clinical narratives into concise, actionable insights, supporting clinician workflows and reducing cognitive overload [2]. Babu and Boddu (2024) demonstrate this evolution by developing a BERT-based medical chatbot that achieves high accuracy and contextual understanding, significantly improving healthcare communication. Their work highlights the shift from basic pattern matching to advanced deep learning techniques in NLP-driven chatbots [11].

These diverse applications reveal several critical insights about NLP in healthcare. First, applications that incorporate contextual understanding are essential for accurately interpreting complex clinical language, particularly deep learning architectures like Bi-LSTMs and transformers. Secondly, domain adaptation and its integration with existing medical knowledge leads to a significant boost in the performance of NLP systems. Third, the fusion of NLP outputs with other data modalities, such as IoT sensor data, enables richer, personalized healthcare solutions. Fourth will be the automation through NLP. It enhances clinical efficiency and patient safety by reducing manual workload and minimizing errors. Last is the

conversational AI, which exemplifies how NLP can extend healthcare delivery beyond traditional settings and improve access and engagement.

The digital sector will continue to evolve, and this kind of data will be generated even more than before. The format of PDFs will remain the same. But it is also true that their static nature resists the enhancement of their capability for interactive and dynamic information access. An intelligent system that couple language interpretation with secure local data deployment is required. It offers an essential step forward. In this context, this research proposes a strong approach to managing, querying, and retrieving information using modern AI tools with security and usability.

Therefore, the integration of NLP into healthcare informatics and conversational AI platforms has already demonstrated substantial benefits, ranging from improved adverse event detection to enhanced patient communication.

1.2 IMPORTANCE OF SECURE DOCUMENT RETRIEVAL

In sensitive environments, including hospitals, courts, academic institutions, and banks, we can't keep data privacy as an option. It is a mandatory notion and regulated by rules such as GDPR (General Data Protection Regulation), HIPAA (Health Insurance Portability and Accountability Act), and other national or organizational data protection laws. Breaking this kind of law can lead to serious consequences. This ranges from legal penalties to reputational damage. This results in the construction of a system to retrieve data from such confidential documents that must ensure that no data leaves the secured system of the organization.

There is an increase in the use of cloud computing. It has given critical security challenges, especially concerning information retrieval from remote servers. Here, sensitive data may be exposed to unauthorized parties or malicious attacks. It helps in ensuring that documents are retrieved securely without compromising their contents or user privacy. This is essential for protecting intellectual property, personal information, financial data, and healthcare records. Cloud providers may offer basic encryption mechanisms, but retrieval methods often require interaction with encrypted data, leading to the need for specialized retrieval techniques that can operate securely on encrypted content [12].

Confidential documents require secure retrieval. This ensures that such information is not exposed to unauthorized personnel during transmission. Organizations should keep internal

access control, which ensures that even within the organization, only those with appropriate roles can have access to the information.

Another critical aspect of secure document retrieval is the efficiency and accuracy of retrieval mechanisms. This aspect must not be sacrificed for security. Effective retrieval systems must ensure that users can obtain the most relevant documents properly even if the operation is within encrypted environments. So, document clustering has been widely used as a preprocessing step. This improves retrieval performance. Users can navigate large datasets more efficiently by arranging documents into semantically related groups. This reduces the time and computational resources needed to locate relevant content. Handa et al. demonstrated that secure clustering techniques can be integrated with encryption schemes to preserve data privacy while enhancing retrieval speed and precision [13].

Methods like clustering-based retrieval also support solutions that are adaptable in distributed cloud systems. Data is spread across multiple servers in large-scale deployments, and secure clustering enables hierarchical search operations and distributed query handling. This improves security, which is done by isolating sensitive information in protected clusters, and performance, as the system avoids querying irrelevant or unrelated data segments [13].

A secure document retrieval system becomes crucial for protecting intellectual property, trade secrets, and confidential communications. With the increase in cyber threats and data sovereignty, organizations are seeking such models that allow access to the data without any threat to the security. The surety that organizations remain compliant with legal standards is a necessity. This can be done by local processing of data while having complete control over access rights and auditability.

The risks, such as third-party involvement or cloud-based services, have also increased in recent years. From data leaks caused by misconfigured cloud storage to serious cyberattacks, the vulnerabilities in remote services mention the need for local and isolated ideas. Secure document retrieval systems eliminate the need for such exposure. This is done by ensuring all data remains within the controlled environment.

Secure document retrieval supports operational efficiency as well as regulatory and security concerns. Organization can accelerate workflows, which reduce administrative overhead by making it less time-consuming to locate essential information and improve service deployment. For example, a lawyer can search for previous cases easily, or a doctor can find a patient's medical history quickly with the help of such a system.

The concept of secure document retrieval extends to multimedia data such as images and videos. These content-based retrieval techniques are crucial in sectors like healthcare and security surveillance. Li et al. proposed a secure content-based image retrieval system where the confidentiality of encryption keys is preserved. The method allows encrypted image features to be compared. It is then retrieved without disclosing the actual image or its descriptors. This is particularly important in medical imaging. In that case, the patient's confidentiality is legally protected [14].

There is a need for secure retrieval. This is further emphasized in commercial and industrial applications where product information and intellectual property are stored in the cloud. Zhao and Zeng proposed an efficient and secure product information retrieval system. The system ensures both data protection and rapid access. Their system integrates secure indexing and hash-based search verification. This guarantees data authenticity and shields proprietary data from tampering or unauthorized access [15]. The commercial implications of secure document retrieval are vast, which means that the companies can collaborate globally while maintaining compliance with data protection regulations and maintaining a competitive edge.

A foundational concept in secure document retrieval is information retrieval theory. The theory highlights the development of algorithms for searching documents, indexing, and ranking. Traditional IR techniques focus on the relevance and effectiveness of the retrieval. Secure IR considers additional constraints such as query privacy, data encryption, and access control. Roshdi and Roohparvar showed a comprehensive overview of IR techniques. This highlighted the way these methods are being adapted to modern security needs. Their work emphasizes the core objectives of IR, which are accuracy, recall, and precision. They must be redefined in secure environments to include confidentiality and resistance to adversarial attacks [16].

From this we can observe that secure document retrieval is a technical enhancement. It is also a critical requirement in modern information systems. The need for secure, scalable, and efficient retrieval methods becomes increasingly urgent as data continues to migrate to the cloud and as privacy regulations grow more stringent. Clustering techniques enhance retrieval accuracy without compromising privacy [13]. This is done to encrypt search frameworks for cloud file systems [12] and domain-specific solutions in multimedia and commercial applications [14][15]. The field has matured to meet real-world demands.

The foundational theories in information retrieval are being re-examined through the lens of security, ensuring that future retrieval systems uphold both performance and protection [16].

Organizations prioritize secure document retrieval to reduce their exposure to cyber threats. These organizations also gain a competitive advantage by maintaining trust, compliance, and operational excellence in a digital-first world.

1.3 MOTIVATION FOR USING NLP IN INFORMATION ACCESS

The motivation to use NLP starts from the problems of traditional keyword-based search methods. Such methods depend on exact keywords, which often provide irrelevant or incomplete results. They do not process the context of the search question. The keyword-based systems are difficult to use. These systems are failing in the case where synonyms or other phrases are used.

Improving user experience in digital platforms is one of the primary motivations for leveraging NLP in information access, which is particularly important in e-services. Ray and Bala realized that understanding what motivates users through NLP usage. Their study employed NLP techniques to analyze textual feedback and user behavior. This reveals the motivational factors that influence user retention, such as convenience, responsiveness, and perceived control [17].

NLP also plays an essential role in eLearning. NLP was used to search the essentials like discussion forums, survey responses, and feedback comments. This shows the issues of lack of personalization, insufficient guidance, and content overload [18]. Educational institutions can design more intuitive interfaces, adaptive learning paths, and responsive help systems by understanding these barriers through natural language data. These promote deeper engagement and information retention. Thus, NLP becomes a tool for content navigation as well as a driver for strategic platform design.

The rise of smart cities increased the use of NLP in information access. The reason is the need to process massive volumes of citizen-generated data from social media, sensors, and public communication platforms. Tyagi and Bhushan highlighted how NLP serves as a foundational component in the infrastructure of an ideal city. This enables the systems to parse real-time data for insights on service demand, public sentiment, and emergency events [19]. Smart city applications require a lot of information. This information is very important to monitor pollution, optimize resource allocation, manage traffic, and improve citizen engagement. NLP enables the given applications by extracting actionable information from unstructured text data. The source for such data can be social media posts, complaint forms, and voice commands. The motivation here is both operational and civic [19].

A compelling motivation for using NLP in information access can be security and trust management. This is particularly true in systems like blockchain. Shahbazi and Byun explored the integration of NLP with blockchain. In their framework, NLP algorithms analyze event descriptions, logs, and reports to detect anomalies and verify the legitimacy of transactions or activities. For example, if a news article reports a cyberattack or a natural disaster, NLP systems can automatically flag the event and trigger verification protocols on the blockchain [20].

The AI technologies, such as machine learning, speech recognition, and computer vision, are prominent. The main example that can be observed is that smart surveillance systems use NLP. NLP combined with machine learning can automate document summarization, policy compliance checks, and contract analysis in the corporate sector, freeing human workers for higher-value tasks [20]. This interdisciplinary synergy enhances the utility of NLP.

NLP changes this pattern by enabling the systems to analyze the structure and intent behind a query. It permits more natural interaction, where users can ask queries in normal human language rather than finding the proper keywords used in documents. The increasing accessibility of the systems is happening through platforms like HuggingFace Transformers. This has made it easier to showcase high-quality NLP capabilities.

In this project, NLP is integrated with semantic embeddings and vector search to give context-aware responses. Embedding models convert text into high-dimensional vectors. These models understand meaning, which enables the system to find relevant answers even when different words are assembled in the query.

In addition, chatbots are conversational in nature, which ensures users explore documents more intuitively. Users can simply put in their queries instead of searching through. This user-friendly approach enables accessibility, particularly for non-technical users, and shows possibilities for engaging such systems in educational sectors, administrative portals, and support systems.

NLP is a good choice for document retrieval. Developers can work on pre-trained models for domain-specific language. This can be done without collecting an extensive database. Thus makes the technology more approachable. NLP acts as a bridge between humans and unorganized digital data, which transforms passive storage into interactive knowledge content.

1.4 RESEARCH OBJECTIVE

The goal of this thesis is to develop and implement a secure chatbot in which NLP is used. It must provide natural language querying and easy retrieval from confidential PDF documents. This system aims to enable the privacy, efficiency, and accuracy of document search processes in sensitive environments. The research objectives include

- To build a program that collects the data and preprocesses text obtained from PDF files using some reliable processing tools.
- To tokenize long documents into smaller chunks for efficient retrieval.
- To generate embeddings related to the content for each segment using sentence transformer systems.
- To implement a vector-based retrieval system, which can be done by using FAISS to provide user queries with relevant text.
- To include a generative AI model. The appropriate model would be Mistral-7B for creating human-like answers based on the retrieved information.
- To ensure that all components work locally. This means that there must be no sharing of any information to cloud servers.
- To create a user-friendly platform that provides easy interaction with the chatbot.
- To analyze the system's performance. This can be executed by considering response time, accuracy, relevance, and security.
- To support approachability and adaptability. This is to ensure future modifications to multilingual support and other improvements.

The objectives also include evaluating the ethical and legal implications of using such technologies in confidential environments and making sure that the proposed system matches with global best practices for data handling and user privacy.

1.5 RESEARCH QUESTIONS

This study is led by several research questions formed to investigate both the theoretical and practical dimensions of designing a secure, NLP-powered document retrieval system. The questions are given below:

- How can NLP be effectively used to retrieve meaningful and relevant information from unstructured PDF documents? This question analyzes the linguistic capabilities of NLP.

It observes how NLP understands document context and responds to queries with factual relevance.

- What are the best practices for tokenizing all the received text to ensure retrieval accuracy? This includes methods of sentence tokenization, chunk sizing, and embedding techniques that best store context and coherence.
- How can FAISS and semantic embeddings be used for secure, high-performance vector search? This evaluates the role of indexing and vector similarity in creating a fast, scalable, and secure retrieval system.
- What are the relationships between retrieval accuracy, system performance, and security in a local deployment? This question shows the balance between accuracy of results, performance speed, and the security limitations.
- How can the engagement of a generative language model enhance the analysis and usefulness of retrieved data? This explains how large language models like Mistral-7B can improve the quality and natural behavior of the chatbot's answers.
- How can modular system design help extensibility and adaptability among various organizational databases? This involves the structural considerations. These considerations allow easy-to-understand updates, domain-specific adaptations, and integration into existing enterprise architecture.
- What ethical and privacy considerations must be approached when AI is used in secure environments? This is crucial to knowing the societal and organizational responsibilities alongside the use of intelligent systems.

The above questions will be answered in the provided study.

1.6 SCOPE AND LIMITATIONS

Natural Language Processing (NLP) can be seen as the foundation of artificial intelligence. This enables the machines to process, interpret, and generate human language. This transformative technology has wide-ranging implications across sectors. The main sectors are healthcare, education, business, and law. In this case, vast amounts of textual data must be interpreted efficiently. NLP is constrained by linguistic, ethical, and technical limitations. This shapes its development and deployment. Its capabilities have expanded rapidly with advancements in machine learning and data availability. This discussion explores the scope as well as the boundaries of NLP with reference to current research and practical challenges.

The scope of NLP is both broad and evolving. Its applications range from basic text processing to advanced linguistic modeling. One of the most significant strengths is automating language-based tasks that traditionally required human involvement. These include information extraction, machine translation, sentiment analysis, text summarization, and conversational interfaces like chatbots and virtual assistants. As Chowdhury and Nath explain about NLP. NLP enables computers to process natural language in a way that mimics human understanding. This allows for more intelligent decision-making systems across a variety of industries [21].

A particularly promising domain is humanitarian action. NLP systems are being employed to analyze large-scale crisis data from social media posts to field reports to identify urgent needs and direct resources more effectively. Rocca et al. (2023) explain how NLP tools have supported disaster relief. He did that by predicting population movements, detecting health concerns, and filtering misinformation in real-time scenarios. The ability of NLP is to extract structured insights from chaotic textual streams during emergencies. This reveals its growing strategic value [22].

NLP also supports scientific and technical innovation in addition to crisis response by aiding in the understanding and disambiguation of complex linguistic structures. This capability is essential in legal and regulatory contexts. The terminology is dense and domain-specific. However, the application of NLP to legal text is not without significant ethical concerns. According to Tsarapatsanis and Aletras in 2021, NLP's deployment raises critical issues related to fairness, bias, and accountability, while NLP can assist in legal research and predictive modeling. This happens especially when it is used in areas like sentencing or policy interpretation [23].

The ongoing development of NLP also brings attention to long-standing linguistic challenges such as ambiguity. Yadav et al. (2021) highlight the multifaceted nature of ambiguity in NLP. They noted that models can struggle with lexical, syntactic, and semantic ambiguities. This is especially true when trained on insufficiently diverse or context-poor datasets. For example, the word "bank" may refer to a financial institution or the side of a river, and without proper disambiguation, NLP systems may produce incorrect outputs that mislead downstream tasks [24].

The scope of this research is based on the design of a prototype model. It analyzes English-language PDF documents. It also functions entirely within a secure, local environment. The

chatbot is developed to retrieve semantically relevant document tokens, process structured and unstructured textual data, and generate natural language responses. Focus is placed on generating a modular structure. It can be taken by various sectors such as healthcare, education, law, and finance.

The system consists of the full NLP program, from document evaluation and preprocessing to semantic indexing and natural language response generation. The technologies involved pdfplumber for text extraction. It also uses NLTK for tokenization and sentence transformers for embedding. FAISS is used for vector similarity search, and the Mistral-7B model is used for generative answers.

However, the study has some limitations:

- Text-only Focus: The system for now does not deal with images. It does not analyze tables or other non-textual content present in PDFs.
- Language Limitation: The prototype is limited to English documents, with no setups for multilingual inputs.
- Resource-Intensive Models: The use of large generative models may not be possible. The available systems are on resource-constrained devices.
- No Voice or Real-Time Interface: Interaction is restricted to typed questions. There is no system for speech-to-text or real-time conversational UI.
- Data Completeness: Accuracy in retrieval depends on the quality and complete data availability of the original PDF content.
- Limited Evaluation: While main performance metrics will be analyzed, broader user studies and long-term deployment are out of scope.

Further, limitations in controlling contradiction and speculation also give rise to prominent issues. Misinterpreting a phrase like “no evidence of disease” or “the defendant might have intended” in domains like medicine and law can lead to serious consequences. Mahany et al. (2022) emphasize that current systems often fail to identify or correctly interpret such linguistic cues. This leads to inaccurate outcomes in text classification and information retrieval tasks [25]. These nuanced elements of language still pose difficulties despite recent advances in neural models. This is particularly true in non-English and low-resource languages.

Another significant limitation arises from the bias in NLP systems. This limitation is due to the data on which they are trained. Language models often learn social and cultural biases present in their training corpora. This includes racial bias, gender stereotypes, or political leanings. This may be inadvertently reinforced by NLP outputs. For instance, if a model is disproportionately trained on Western literature or internet forums, it might display partiality in its results when applied to multicultural settings. This is especially problematic when NLP is used in sensitive applications like hiring, healthcare triage, or law enforcement [23].

There are technical constraints related to generalizability in addition to ethical concerns. NLP systems trained in one domain or language may not transfer well to another. The main example is a model trained on English legal documents may not perform reliably when applied to medical records in French or Hindi. This issue of domain shift restricts the adaptability and universality of NLP tools. Moreover, resource-intensive training and infrastructure requirements restrict access to cutting-edge models. This often leaves smaller institutions or organizations in developing regions at a disadvantage [20].

One of the more subtle but critical limitations is the lack of transparency in NLP models. This is particularly true of deep learning architectures like transformers. These models, while highly accurate in many tasks, often function as black boxes, making predictions without offering interpretable rationales. The major concern is opacity in high-stakes environments such as legal decision-making. Here, explainability is essential for accountability and regulatory compliance. Tsarapatsanis and Aletras (2021) argue on this issue. This deployment of NLP in legal systems without interpretability mechanisms may undermine democratic and procedural fairness [23].

Despite these limitations, the system is developed by considering extensibility, allowing for future enhancements such as optical character recognition (OCR), multilingual models, and role-based access control.

1.7 THESIS STRUCTURE OVERVIEW

This thesis is divided into seven chapters, each addressing an essential component of the research, design, implementation, and evaluation of the presented chatbot system:

Chapter 1: Introduction — Provides an outline of the existence of the issue. This includes background, motivations, and research questions, which present the basis for the study.

Chapter 2: Literature Review — Reviews existing literature and technologies related to NLP-based information retrieval, secure systems, semantic search, and chatbot interfaces. This chapter explores research gaps and positions the current study within the larger academic and industrial context.

Chapter 3: System Design and Methodology — Explains the system architecture, data flow, component modules, and the reason behind the chosen technology. This consists of discussions on preprocessing techniques, model selection, and security protocols.

Chapter 4: Implementation — Evaluates the practical development of the system, presenting code structure, toolkits used, configuration details, and implementation challenges. This chapter highlights how different technologies were integrated into a cohesive system.

Chapter 5: Evaluation and Results — Shows an empirical evaluation of system performance, including metrics such as response accuracy, retrieval speed, and system security. Comparisons with basic approaches are provided where applicable.

Chapter 6: Discussion — Gives essential insights into the results, analyzes strengths and weaknesses of the system, and discusses larger implications such as ethical considerations and deployment feasibility.

Chapter 7: Conclusion and Future Work — Summarizes the findings, reiterates contributions, and guides towards potential directions for future improvements, such as real-time chat UIs, multilingual processing, and cloud-secure deployments.

Together, these chapters build a comprehensive narrative that walks the reader from problem identification to the realization of a secure, intelligent document retrieval system. This structure ensures clarity, logical progression, and alignment with both academic standards and real-world applicability.

2

Literature Review

2.1 NATURAL LANGUAGE PROCESSING IN INFORMATION SYSTEMS

Natural Language Processing (NLP) is considered a great development in modern information systems. NLP enables enhanced data processing, better user interaction, and the automation of complex analytical tasks in the context of information systems. It is evident that there is exponential growth of unstructured data. It ranges from emails and reports to scientific papers and regulatory documents. The role of NLP has become more crucial than ever.

Evolution of NLP in Information Systems

The use of language-based systems within information systems is not a new concept. The systems that existed before heavily relied on rule-based engines and keyword matching. These systems were able to process structured queries, and they often failed to get contextual meaning, ambiguity, and intent. Liu et al. [26] observed a comprehensive roadmap that identifies NLP as a central research pillar in progressing intelligent information systems. This work highlights the way NLP methodologies enhance semantic search, context-aware querying, and adaptive interfaces.

Arslan et al. [27] proposed that NLP plays an important role in digital transformation in management information systems (MIS). It is able to enhance data accessibility and user experience. They argue that traditional information systems are now being enhanced with NLP to provide dynamic and conversation-like user experiences, which once relied on rigid form-based inputs. These systems reduce the technical barrier for users. It also improves the efficiency and relevance of information retrieval.

Applications of NLP in Document and Domain-Specific Information Systems

There are various applications of NLP in domain-specific contexts. This demonstrates that NLP is adaptable and precise when tailored correctly. One notable domain of this is chemical information systems. He et al. [28] has depicted NLP as a highly effective system to extract data from complex scientific patents. They have presented a study in the CHEMU 2020

competition that demonstrated that pre-trained language models could successfully identify chemical compounds, dosages, and reactions from dense and technical patent documents. This shows how NLP can turn unstructured domain-specific content into structured and actionable knowledge, which is essential for R&D and compliance functions.

In the thesis, the goal is to build a secure NLP-powered chatbot. It should be able to retrieve appropriate answers from confidential documents. The approach integrates PDF analysis, sentence tokenization, semantic embeddings, and generative modelling. This enables the formation of an intelligent system that provides human-like responses. This system processes raw documents. This helps users to interact with them using natural language. It offers answers, context, and clarity. It serves as an example of how NLP techniques can be applied within an isolated and secure environment. It reinforces the broader trend of deploying NLP for domain-specific tasks.

Natural Language Systems and End-User Experience

The evolution of natural language interfaces (NLIs) is one of the most significant recent developments in NLP for information systems. These systems allow users to search engines, query databases, or document repositories using English instead of structured query languages. Majhadi and Machkour [29] provide a historical overview of natural language interfaces for databases (NLIDBs). It analyses how they have evolved from early syntax-based systems to current transformer-powered solutions. Different researches underscore the importance of adaptability, robustness, and domain awareness in NLP interfaces.

There is a rise of transformer-based models such as BERT, GPT, and Mistral. This has improved the quality of NLIs drastically. These models understand the words in a query and the intent behind them. By introducing NLP into information systems, organizations can reduce the learning curve for non-technical users and democratize data access. They can also improve overall operational agility.

NLP and the Future of Intelligent Information Systems

The complexity of data is continuing to grow. NLP plays an important role in the evolution of intelligent information systems. Liu et al. [26] proposed that the next generation of IS will rely heavily on NLP for information retrieval as well as for knowledge representation, summarization, and decision support. Similarly, Arslan et al. [27] said that NLP will become

the primary way of interaction for future MIS. This allows users to evaluate data without the need to navigate complex dashboards.

This vision deploys a local, secure, and domain-aware chatbot. The system shows how NLP can support high-level cognitive tasks. This can be done by processing documents into semantically indexed chunks and utilizing generative AI for answering queries. This aligns with broader trends in IS. The integration of NLP is an enhancement and a necessity for competitive digital transformation.

NLP will soon be used for more developments in on-device processing, privacy-preserving models, and federated learning. Systems will be able to offer intelligent functionality while respecting data ownership and confidentiality. This is particularly relevant for sectors like law and healthcare, where compliance with standards such as HIPAA or GDPR is mandatory.

2.2 EVOLUTION OF QUESTION-ANSWERING CHATBOTS

The evolution of question-answering (QA) chatbots is one of the most dynamic developments in the field of Natural Language Processing (NLP). QA chatbot systems are capable of understanding user queries and delivering contextually accurate responses. They have progressed from rule-based scripts to sophisticated and multi-layered conversational agents, which are powered by generative models. Organizations across various sectors increasingly adopt AI-driven interfaces for information access. The technological evolution of QA chatbots becomes essential. This happens especially in secure, data-sensitive domains such as healthcare, finance, and legal document retrieval.

Rule-Based, Statistical, and Retrieval-Based Systems

The earliest generation of chatbots was rule-based. It depends on fixed templates and pattern-based ideas. These systems simulated conversation through pre-programmed responses without true understanding of language, such as ELIZA and PARRY. Rule-based chatbots demonstrated the feasibility of human-computer interaction through language.

With the advancements in statistical methods in the late 1990s and early 2000s, chatbots began to incorporate probabilistic models to improve flexibility and relevance. These kinds of systems used techniques such as Hidden Markov Models (HMMs) and Naive Bayes classifiers. This is done to determine the most appropriate response based on word frequency and pattern odds.

Retrieval-based QA systems took this a step forward, which is done by incorporating information retrieval techniques. These systems marked the transition from scripted responses to dynamic information extraction, though they still lacked contextual depth.

The Introduction of Deep Learning and Multi-Layered Architectures

The involvement of deep learning techniques was a significant turning point in chatbot development. Long Short-Term Memory (LSTM) networks and sequence-to-sequence models became capable of generating original responses based on context and past dialogue with the rise of recurrent neural networks (RNNs). These architectures introduced memory and temporal awareness into conversational AI. This significantly improves naturalness and coherence.

Makhkamova and Kim [30] explored the architecture of multi-layered chatbot services by using a conversation history-based caching mechanism. The study showed that chatbots equipped with memory and hierarchical design. This could personalize responses, manage long-term user interactions, and reduce latency in retrieving accurate answers. This innovation is highly relevant in enterprise-grade chatbots.

Transformer-Based and Generative Chatbots

The real revolution in chatbot technology started when transformer based models were introduced. Transformer architectures process entire sequences simultaneously using self-attention mechanisms. This significantly enhances their ability to model complex dependencies in language. The release of models like BERT, GPT, and T5 set a new standard in machine understanding and generation.

Al-Amin et al. [31] provide a comprehensive historical overview of generative AI chatbots. This traces the field from rule-based systems to contemporary transformer-based architectures. Their study emphasizes that the evolution of generative models has enabled chatbots to engage in open-domain conversations. This offers explanations, and it also summarizes complex information and capabilities that are highly desirable in confidential document retrieval applications.

One of the key strengths of generative chatbots is their ability to synthesize multiple sources of information to produce appropriate responses. This is particularly useful in situations where

the answer is distributed across several document sections. The chatbot developed in this thesis exhibits this capability by integrating a retrieval-augmented generation (RAG) pipeline, where the Mistral-7B model generates answers grounded in the retrieved document context.

Domain-Specific QA Chatbots

Mansurova et al. [32] developed a QA chatbot for the blockchain domain. This study depicts the importance of domain knowledge as well as contextual embeddings. Their system used NLP to interpret user queries. It interprets knowledge from a curated blockchain database. The chatbot's architecture included modules for semantic analysis, knowledge base retrieval, and response synthesis. This study reinforces the value of customizing chatbot pipelines for particular domains. The principle presented in this thesis is where the chatbot is trained and evaluated on confidential documents unique to a given organization.

Comparative Evaluation of Chatbot Algorithms

As chatbot architectures have become popular, there is the need for systematic comparison of their capabilities. A comparative review of machine learning algorithms used in chatbot development [33] was conducted. Their findings revealed that rule-based and retrieval-based models are more efficient and easier to implement, but they lack the adaptability and contextual understanding offered by generative models.

According to their evaluation, transformer-based models excelled traditional methods in metrics such as fluency, relevance, and user satisfaction. This was particularly true in open-domain and knowledge-intensive tasks. However, they also noted that these models come with higher computational demands. They require substantial fine-tuning to avoid hallucination and ensure factual correctness.

The use of a local FAISS index for retrieval and the Mistral-7B model for generation reflects a strategic choice to leverage the strengths of both approaches.

Chatbots in Secure and Confidential Environments

The deployment of chatbot technologies in secure environments presents unique challenges and opportunities. It is known that most commercial chatbots rely on cloud-based infrastructures. But it has been shown that organizations dealing with confidential or regulated data require local and isolated solutions that can guarantee privacy.

In such contexts, generative chatbots must be optimized for local deployment. This will ensure that user data never leaves the internal network. This includes lightweight models, on-device inference, and encrypted storage. The multi-layered architecture was proposed by Makhkamova and Kim [30]. It provides a useful blueprint for designing such systems.

The chatbot implemented in this thesis shows that from preprocessing and semantic indexing to query interpretation and response generation, all operations are performed locally. This ensures compliance with data privacy regulations and aligns with enterprise-grade deployment standards.

2.3 SEMANTIC SEARCH AND DOCUMENT EMBEDDING TECHNIQUES

Users no longer seek documents containing exact words, but they seek meaning. Semantic search represents a shift from lexical to conceptual information retrieval. This approach enables systems to retrieve documents based on their underlying semantics, regardless of whether the query uses the exact terms found in the source text.

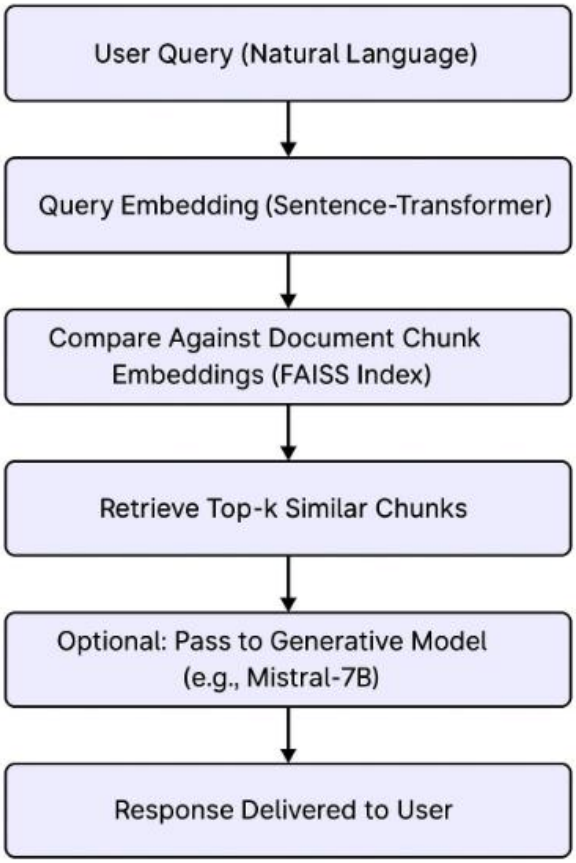

Figure-2.1: Query Embedding Using Semantic Search

From Keyword Matching to Semantic Retrieval

Conventional search engines use keyword-based techniques. These rely heavily on term frequency-inverse document frequency (TF-IDF), Boolean operators, or BM25 algorithms. While these methods are computationally efficient, they suffer from several limitations. The limitations include the inability to recognize synonyms, handle polysemy, or understand contextual relevance. As a result, they often return documents that contain the query terms but not necessarily the desired information.

Semantic search overcomes these limitations, which is done by converting both queries and documents into dense vector representations using embedding techniques. These embeddings capture semantic relationships between words, phrases, or entire documents, which allows the measurement of conceptual similarity in high-dimensional space. Ji et al. [34] provide an appropriate idea of this phenomenon. The study illustrates how semantically similar documents cluster together in vector space. Their work demonstrates the value of embedding-based methods for interactive exploration and improved information retrieval accuracy.

In the context of the present thesis, semantic search enables users to query large volumes of confidential documents in natural language and receive answers that are contextually relevant, even if the wording differs significantly from the source material.

Word and Document Embeddings

Word embeddings are numerical representations of words that preserve semantic relationships. Models such as Word2Vec, GloVe, and FastText are foundational to semantic search. This allows systems to analyze similarity based on data. For example, embeddings enable systems to recognize the semantic relationship between “attorney” and “lawyer,” even if they do not co-occur frequently in collected data.

Document embeddings represent entire sentences, paragraphs, or sometimes documents as single vectors. Sentence transformers are built on top of BERT and its variants. They have enabled high-quality semantic encoding of longer text units. Webler [35] used word embeddings to develop a semantic search engine for tagged artworks. The work shows that even short text data can be effectively matched using embedding similarity. His study highlights the power of embeddings, which supports exploratory and concept-based search.

These techniques are highly applicable in document retrieval systems designed for enterprises, where query phrasing may vary significantly among users and documents are often domain-specific.

Vector-Based Search and Neural Indexing

The semantic search is used in constructing a vector space index in which embeddings of all documents are stored. When a user has a query, the embedding of that query is computed and matched against this index by using similarity metrics like cosine similarity or Euclidean distance. The most similar vectors are obtained as results.

Monir et al. [36] introduced VectorSearch as an appropriate semantic retrieval system. It combines document embeddings with an efficient indexing mechanism. Their system outperformed traditional keyword-based engines in both accuracy and response time. VectorSearch represents the state of the art in information retrieval. This is done by integrating deep semantic models with high-performance vector indexes like FAISS or Annoy.

Enhancing Similarity Metrics

Yan et al. [37] explored various similarity metrics related to word embeddings and found that the choice of distance metric can significantly influence the quality of retrieved results. They proposed improvements to existing similarity measures, which helps in better analysis of semantic nuance, particularly in domain-specific applications.

These findings are important for evaluating document retrieval systems. This is especially true when operating in highly specialized or sensitive domains. It customizes similarity functions and adjusts the dimensionality of embeddings. It also optimizes chunk sizes.

Relevance to Confidential Document Retrieval

In sensitive scenarios, users often need to retrieve information based on concepts or implications rather than exact wordings. A semantic search engine would correctly identify this match, but a keyword-based engine might not do so.

Additionally, embedding-based indexing allows the retrieval to be compact, encrypted, and efficient. This is important in a secure system where documents are stored locally and cannot be shared with third-party cloud providers.

2.4 DATA SECURITY AND CONFIDENTIALITY CHALLENGES

As the sensitive digital data continues to grow everywhere, ensuring data security and maintaining confidentiality have become critical concerns. Secure document retrieval systems are especially using natural language interfaces. Therefore, it must be designed with a rigorous understanding of the threats and mitigation strategies associated with data privacy.

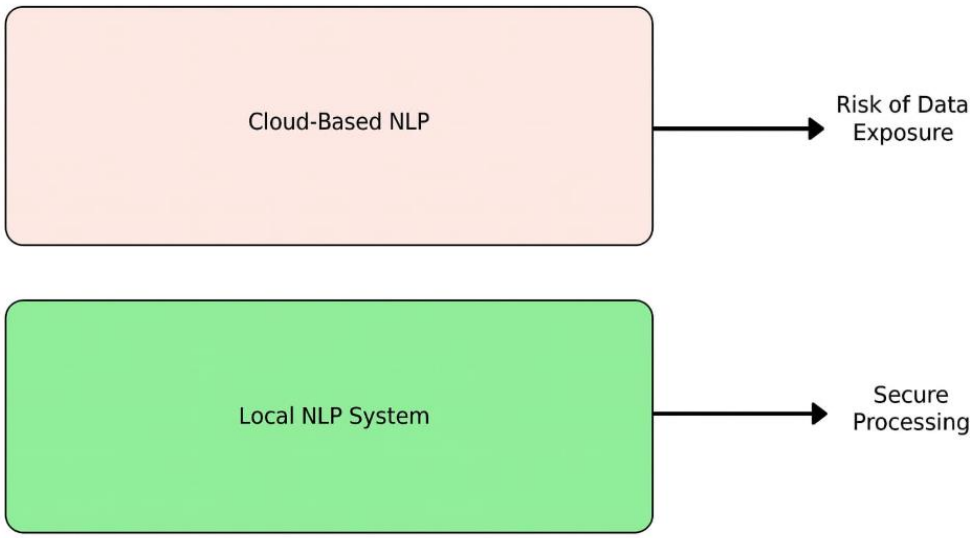

Figure-2.2: NLP ensuring Data Security and Confidentiality

Cloud-Based Systems and Security Vulnerabilities

Most of the modern data management platforms rely on cloud computing. This happens due to its availability, adaptability, and cost efficiency. However, cloud infrastructures are inherently exposed to several security threats. These threats include insider attacks, service vulnerabilities, data breaches, and unauthorized access. Rao and Selvamani [38] represent these challenges and emphasize the lack of direct user control over data once it is outsourced to third-party providers. They identify multi-tenancy, data segregation issues, and unsecured APIs as key vulnerabilities that could compromise sensitive information.

The cloud-related risks make conventional NLP deployment models unsuitable. Data stored or processed in the cloud could inadvertently be exposed. These threats justify the use of local and offline NLP deployments.

Confidentiality in Economic and Business Domains

The problem of data confidentiality is not exclusive to cloud-based systems. Even within isolated environments, ensuring secure access control and information compartmentalization remains a major challenge. Portovaras et al. [39] highlight the specific risks faced by businesses when they have to handle sensitive economic data. This includes strategic documents, financial reports, and audit trails. They argue that even internal leaks or inadvertent exposure of confidential analyses can have significant legal, reputational, and competitive repercussions.

These concerns are elevated when automated document retrieval systems are used. Chatbot interfaces may unintentionally expose sensitive document contents through query responses without strict access control and encryption protocols. Therefore, NLP-powered document systems must integrate multiple layers of security like encryption at rest and in transit, user authentication, role-based access control, and logging mechanisms to trace queries and responses.

Privacy in Healthcare and Care Coordination

In domains like healthcare, the trade-off between information accessibility and privacy is very sensitive. Patient data must be shared among healthcare professionals for effective diagnosis and treatment. It must also be protected from unauthorized access to preserve individual confidentiality. Ibrahim et al. [40] describe this challenge as a balancing act between care coordination and data protection.

This solves problems like how to enable efficient retrieval while considering the safety of confidential details. The chatbot developed in this thesis addresses this challenge through design choices such as local vector indexing, semantic chunk retrieval, and response grounding.

2.5 ACCESS CONTROL AND AUTHENTICATION IN NLP APPLICATIONS

Natural language processing (NLP) technologies have become increasingly embedded in information systems, which ensures secure access to these systems through effective access control and authentication mechanisms. This has become a critical concern. There are certain environments where sensitive data is retrieved or generated via conversational interfaces, and so it is vital to restrict access based on user roles, attributes, or behavioral traits. The application of NLP in access control models, policy interpretation, and user authentication offers new opportunities for building more adaptive, secure, and intelligent systems.

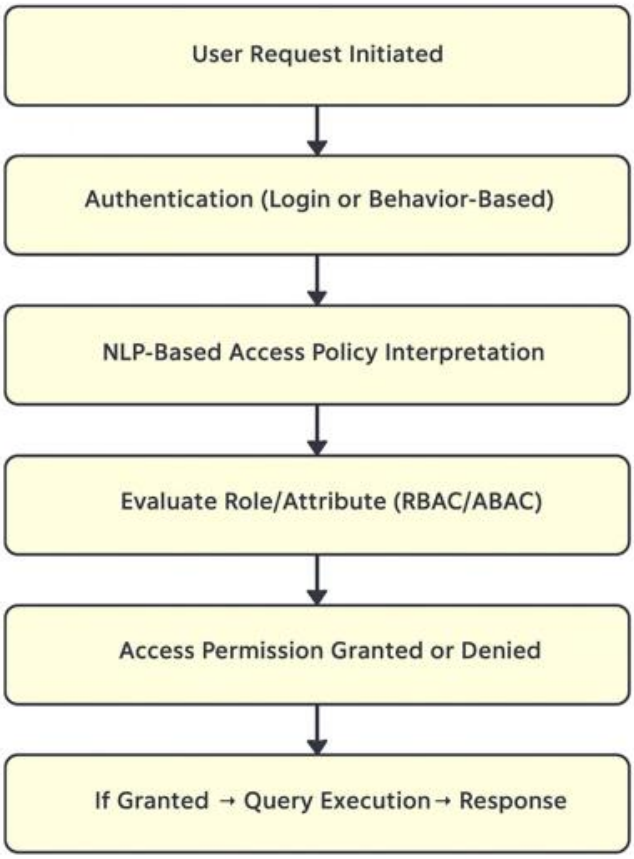

Figure-2.3: Application of NLP in Access Control Models and Authentication

Natural Language Policy Interpretation and Access Control Generation

Access control mechanisms used to rely on explicit rule sets and configurations defined manually by system administrators. These systems often require users to interact with technical policy languages or markup configurations. These configurations are error-prone and difficult to manage at scale.

Jayasundara et al. [41] present a comprehensive systematization of knowledge on translating high-level natural language access requirements into enforceable security policies. The survey covers various policy models. This includes Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), and Discretionary Access Control (DAC). The authors discuss the key technical challenges involved in interpreting and formalizing policy descriptions expressed in natural language. These challenges include resolving ambiguity, identifying logical conditions, extracting roles and permissions, and mapping them to enforceable structures.

This approach is highly relevant in the context of an NLP-powered document retrieval chatbot. The system must be able to infer access rights from natural language descriptors of user roles or document classifications. Automating the process by using NLP enhances scalability and reduces the complexity associated with managing large-scale access configurations across departments or sectors.

Attribute-Based Access Control Models from NLP

Attribute-based access control (ABAC) offers a more flexible approach. They consider a combination of user attributes, resource attributes, environmental conditions, and action types to access permissions.

Abdelgawad et al. [42] explored the synthesis of ABAC models from textual policy inputs. The approach of this study involved parsing natural language descriptions of access rules and converting them into formal, machine-readable access policies. They achieved this using named entity recognition (NER), dependency parsing, and semantic role labeling. This enables the accurate extraction of relevant attributes, constraints, and logical operators.

NLP-Based User Authentication and Behavioral Biometrics

Authentication is the process of verifying a user's identity. It has traditionally relied on static methods such as passwords, tokens, or biometric scans. However, NLP offers new avenues for continuous and context-aware authentication. This is done by analyzing behavioral patterns in user interactions. One such innovation involves the use of mouse dynamics. Mouse dynamics are a user's unique way of moving the mouse. It is used as a biometric trait that can be modeled and interpreted through machine learning and NLP techniques.

Lee et al. [43] investigated this concept by developing an NLP-based user authentication system. The system utilizes mouse movement patterns during natural language tasks. The system captures mouse trajectories, click timing, and acceleration curves while users engage in text input. It uses NLP models to correlate these behaviors with known identity profiles. The result is a passive and continuous authentication mechanism. It enhances security without requiring intrusive inputs.

Integration into Secure NLP Systems

The integration of NLP-based access control and authentication mechanisms into a secure chatbot system requires a systematic architecture. The establishment of identity management

and user roles must be done through secure sign-in protocols. NLP modules can interpret dynamic access policies written in natural language and enforce them in real-time. Finally, behavioral monitoring using NLP or hybrid techniques can provide continuous assurance of user legitimacy.

In this thesis, users can access control, which must operate both at the user-interface layer. This means what questions are allowed or not, and at the document layer, that means what content is retrievable. By using NLP to access permissions and analyze user behavior, the system can enforce context-sensitive security policies, reducing the likelihood of unauthorized access or data leakage.

2.6 REVIEW OF RELATED TOOLS

The implementation of a secure NLP-powered chatbot for confidential document retrieval relies on a range of specialized tools and frameworks. Each component of the system requires robust technologies that can handle large-scale data, maintain contextual relevance, and ensure privacy. This section reviews the core tools and libraries used in the thesis system, focusing on their capabilities, relevance, and integration.

1. pdfplumber – PDF Text Extraction

The first stage in building a document retrieval system is to acquire textual content from raw documents, many of which exist in PDF format. The pdfplumber library is used to extract structured and unstructured text from PDF files while preserving layout and metadata. Unlike simpler parsers, pdfplumber can handle multi-column text, footnotes, and headers with greater accuracy. This makes it particularly suitable for enterprise documents such as reports, contracts, or scientific articles.

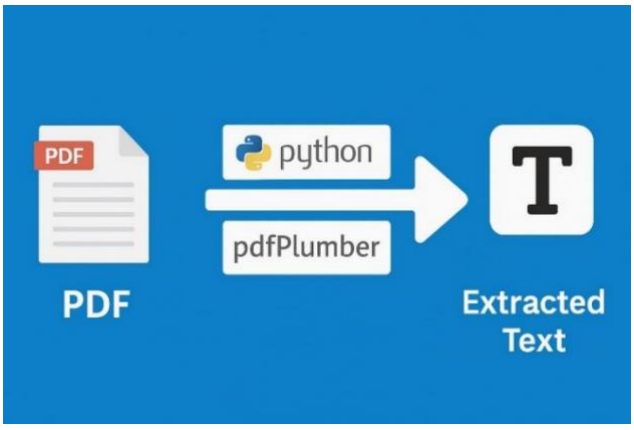

Figure-2.4: Use of pdfplumber to extract text from PDFs [44]

In this thesis, pdfplumber is responsible for ingesting a collection of local PDF files, converting each into a textual representation, and preparing them for further processing.

2. NLTK – Tokenization and Text Normalization

The Natural Language Toolkit (NLTK) is used in the preprocessing pipeline to split large documents into manageable chunks. Tokenization allows the system to preserve semantic coherence while segmenting text into units.

Chunking is essential for accurate retrieval because large documents must be broken down in a way that does not compromise meaning. In the thesis, a fixed-length chunking method is used, which is guided by NLTK's sentence tokenizer to avoid cutting sentences from the middle. This process improves both the retrieval method and the quality of generated responses.

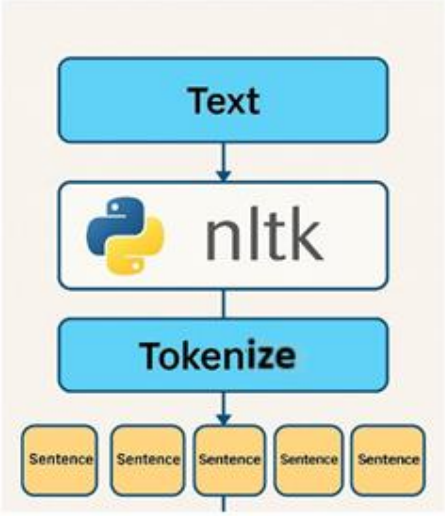

Figure-2.5: Tokenization of text using NLTK

3. Sentence Transformers – Semantic Embeddings

Sentence transformers is a framework that is built upon HuggingFace's Transformer models and PyTorch. This is central to the semantic retrieval capabilities of the system. It allows for converting text of a query and document chunks into dense vector representations that encode meaning.

These sentence embeddings are used to calculate similarity between a user query and document chunks in the vector space. The model in this thesis shows a balance between performance and

speed. It provides robust embeddings suitable for large-scale document indexing and real-time retrieval.

Chandra et al. [45] show that sentence transformers offer a scalable solution for converting documents into high-dimensional embeddings. This can be effectively evaluated using similarity metrics. The pipeline for embedding and retrieval highlights the design choices made in this thesis, which include the decision to preprocess data offline and use compact as well as efficient models.

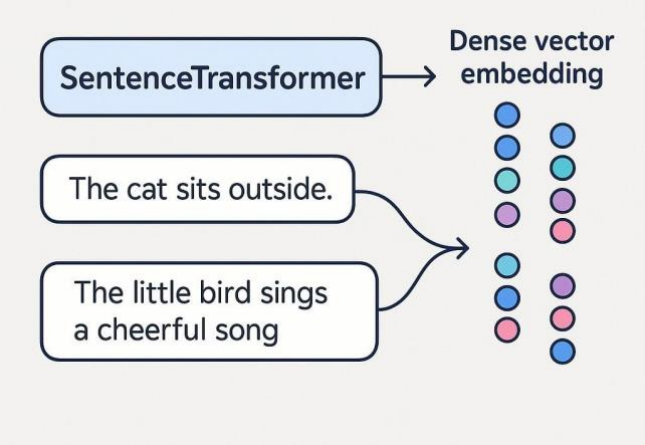

Figure-2.6: Use of Sentence Transformer in Semantic Embeddings

4. FAISS – Vector Indexing and Similarity Search

The Facebook AI Similarity Search (FAISS) library is used for indexing and retrieving vectors based on similarity. It supports approximate nearest neighbor (ANN) search. This allows the system to retrieve the top-k most relevant document chunks for any given query embedding.

FAISS was selected in this project due to its high performance, adaptability, and support for CPU-based operation. This is an important factor for local deployments where GPU resources may be limited. The system uses FAISS to construct an index of all document chunk embeddings. This index can be searched in milliseconds during inference.

Chandra et al. [45] used a distributed FAISS setup in their large-scale architecture for managing document retrievals. This thesis employs a single-node version due to local constraints, although the architectural flexibility provided by FAISS allows future scalability without redesigning the pipeline.

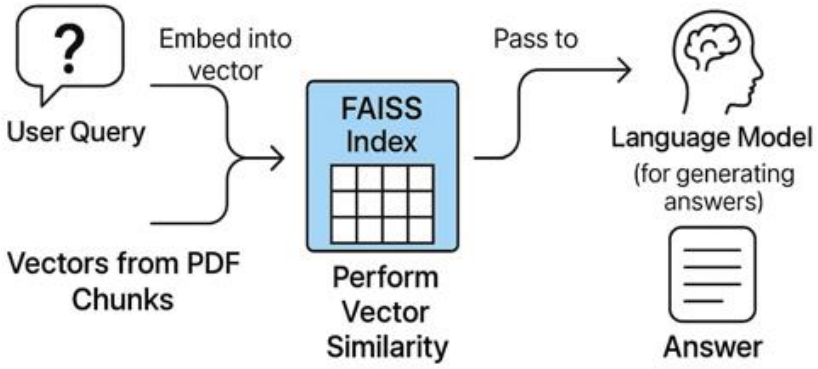

Figure-2.7: Use of FAISS Index to perform vector similarity

5. Mistral-7B – Generative Response Model

Mistral-7B is a modern open-weight language model. It is known for its balance of performance, efficiency, and adaptability. It is used as the generative layer in the chatbot. In this thesis, it is responsible for interpreting user questions, leveraging retrieved context, and producing fluent as well as helpful responses.

Mistral-7B is capable of summarizing, rephrasing, and synthesizing information. This makes it particularly useful when document content is fragmented or indirectly related to the user query.

Some studies [46] highlight the use of similar large language models within retrieval-augmented generation (RAG) frameworks to create AI-based support systems. The architecture combines semantic retrieval with contextual generation and recommendation. This validates the approach taken in this thesis.

The chatbot here follows a similar pattern, which is shown below:

retrieve → augment → generate

This ensures both accuracy and fluency in outputs.

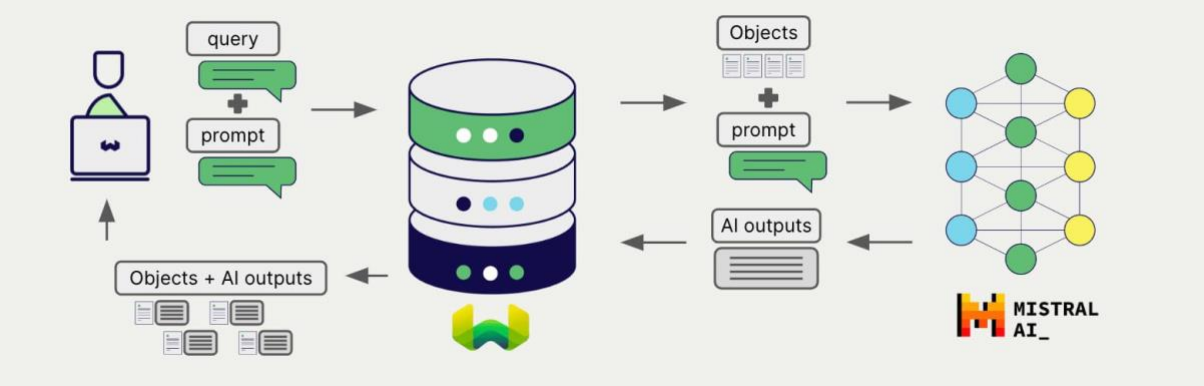

Figure-2.8: Mistral-7B as Generative Response Model [47].

2.7 COMPARITIVE ANALYSIS OF EXISTING SYSTEMS

There is an immense growth in the adoption of Natural Language Processing (NLP) across sectors like healthcare, knowledge management, and customer service, which has led to a wide range of systems. Each system has its own strengths and limitations. Analyzing these systems provides important insights that inform the design of the secure and document-retrieving chatbot proposed in this thesis.

Bose et al. [48] conducted a comparative study on NLP applications in COVID-19 research. It covers tasks such as extraction of information and classification of documents. They showed that transformer-based models like BERT and SciBERT far outperformed earlier rule-based and statistical models. This is particularly true in handling context and domain-specific content. These findings support the thesis and its use of transformer-based embeddings to process complex PDF documents.

Zaoui Seghroucheni et al. [49] examined knowledge management of NLP in enterprise settings. The research found that older models relying on handcrafted features performed poorly on unstructured data. In contrast, models like Sentence-BERT and T5 delivered better scalability and retrieval accuracy. This justifies the thesis's use of sentence transformers to semantically encode documents for effective retrieval.

Chaid et al. [50] compared classical algorithms such as naive Bayes, decision trees, and SVMs with deep learning methods. Traditional models were noted for speed and simplicity but struggled with tasks requiring deep contextual understanding. Deep learning models proved superior in areas like semantic similarity and question answering. This supports the use of neural-based approaches in the chatbot.

Upreti [51] compared conversational models like LSTM, BERT, GPT, and XLNet. The results showed that models like GPT and XLNet generated more relevant and context-aware responses. It is to be noted that it required significant computational resources and careful tuning. These insights influenced the thesis's choice of the Mistral-7B model. This is deployed locally to ensure both performance and data privacy.

The advanced models also have limitations. Transformers require high computational resources and lack transparency in decision-making. It is true that simpler models are more interpretable and lightweight, and they fail to handle semantic nuances effectively. The thesis explains this by using a modular approach, which uses efficient sentence embeddings for retrieval with a lightweight transformer-based model for response generation.

Overall, these comparative studies validate the thesis's architecture and deployment strategy, aligning with best practices while prioritizing data privacy and usability.

2.8 SUMMARY OF RESEARCH GAPS

Despite recent advances, several gaps remain in the field of secure and intelligent document retrieval using NLP. This thesis aims to address the most pressing of these.

Firstly, most current NLP systems operate in cloud environments, raising concerns about privacy and data leakage. This is a concern, especially in sensitive fields like healthcare and finance. Few systems offer secure and offline alternatives. This thesis addresses this gap by implementing a fully local NLP system, which ensures confidential data remains within user control.

The second research gap would be the focus of existing systems on isolated components. This can either be document retrieval or response generation but rarely both in an integrated manner. The thesis presents a solution for this. From document ingestion and preprocessing to semantic search and response generation, it offers a seamless user experience.

Third is that its application to real-world PDF documents remains limited. PDFs can contain inconsistent formatting, unstructured text, and embedded elements that complicate the processing. The thesis introduces a novel and effective solution by combining tools like pdfplumber for extraction with sentence transformers and FAISS for retrieval.

Fourth is a lack of implementation of natural language-based access control mechanisms in chatbot systems. The thesis outlines the integration of these features and allows role-based and secure document access within the conversational interface.

Fifth, retrieval-augmented generation (RAG) systems are often built for open-domain tasks. They rely on cloud APIs. There is limited research on adapting RAG frameworks for secure and domain-specific use. The thesis contributes by building a local RAG system using FAISS and Mistral-7B to enable accurate and private document-based responses.

Lastly, many NLP systems are tested on corrected datasets under ideal conditions. This fails to reflect the noise, variability, and domain-specific challenges found in real data. The thesis system is evaluated using diverse and locally stored PDF files to simulate realistic usage and deployment scenarios.

3

System Design and Methodology

3.1 SYSTEM OVERVIEW AND ARCHITECTURE DIAGRAM

The design of the secure NLP-powered chatbot presented in this thesis integrates a sequence of modular components that collectively enable confidential document retrieval using natural language queries. The system adopts an end-to-end system architecture, which ensures that each module performs a specific role. This ranges from document ingestion to semantic search and response generation. This section provides an overview of the system's architectural design, elaborating on the data flow, components, and interactions that make up the functional core of the chatbot.

3.1.1 System Goals

The primary goal of the system is to help users to analyze a set of sensitive PDF documents and receive contextually relevant and human-like responses. This all should be done without compromising data confidentiality. To meet this objective, the system must:

- Extract and preprocess text from PDF documents securely and confidentially.
- Break down the content into structured and semantically logical chunks.
- Embed the chunks into a high-dimensional vector space.
- Allow semantic evaluation using natural language input.
- Retrieve relevant document segments based on similarity to the asked query.
- Use a generative language model to answer the query using the retrieved content.
- Operate entirely offline to ensure data privacy.

Each of these tasks is performed by a dedicated module. This is integrated into a seamless and extensible architecture.

3.1.2 High-Level Architecture

The system architecture is composed of the following core components:

1. **PDF Ingestion Module** – It uses pdfplumber to extract text content from PDF files, which is stored locally. This ensures that no sensitive data is transmitted externally.

2. **Preprocessing and Chunking Module** – Document is preprocessed using NLP. Then it employs NLTK to tokenize text into sentences. After that, it aggregates them into fixed-length chunks for embedding. This step facilitates more accurate semantic indexing.

3. **Embedding Generation Module** – This module utilizes sentence transformers to convert document chunks into dense vector embeddings. These embeddings preserve the semantic meaning of the text and allow similarity comparisons.

4. **Indexing and Retrieval Module** – FAISS is used to index all embeddings and supports fast search to retrieve the most relevant chunks for a given query.

5. **Query Processing Module** – This module encodes the user's natural language question and then converts it into a vector using the same embedding model. Then, the query vector is matched against the FAISS index, which helps to retrieve similar content.

6. **Contextual Answer Generator** – This answer generator implements Mistral-7B by the use of the HuggingFace transformers system to generate human-like answers. The model takes the retrieved document context and the user's question as input. This is done to generate a coherent and context-aware response.

7. **Security Layer** – It ensures all operations are performed locally. It makes sure that documents, embeddings, and queries never leave the host machine. This layer also supports future extensions such as user authentication and access control.

3.1.3 Architectural Diagram

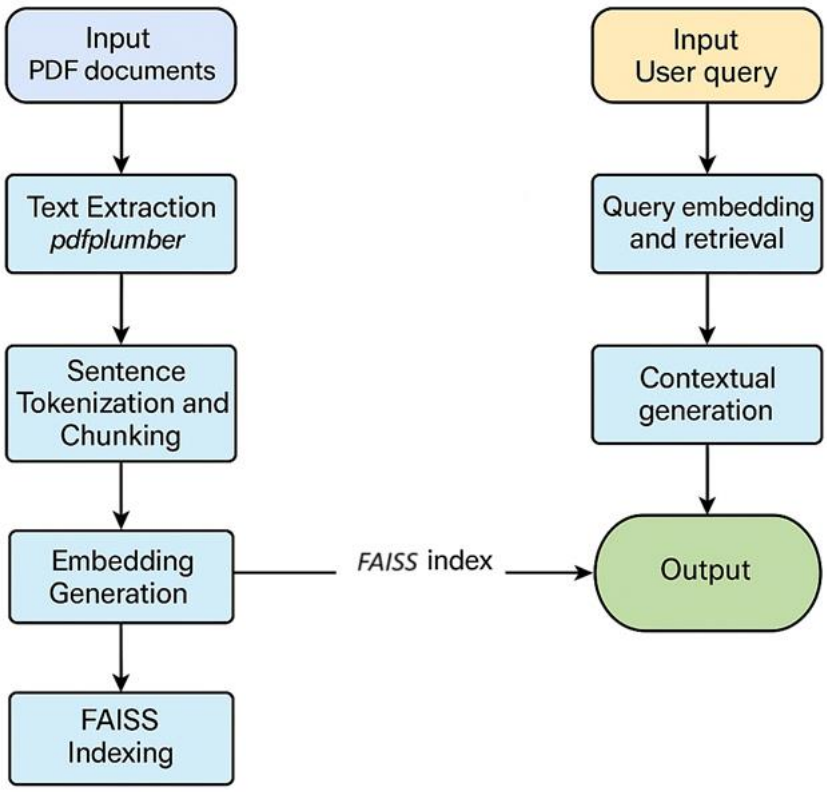

Figure-3.1: Architectural Diagram of the System

The design can be described as follows:

1. **Input Layer** – The system takes a set of PDF documents as input at the start.
2. **Text Extraction** – The documents are resolved using pdfplumber, and then the content is extracted as raw text.
3. **Sentence Tokenization and Chunking** – The raw text is segmented into sentence based chunks using NLTK.
4. **Embedding Generation** – Each chunk is passed to all-MiniLM-L6-v2, which is the sentence transformer model. It is done to obtain its vector representation.
5. **FAISS Indexing** – All chunk embeddings are indexed using FAISS. This helps to create a searchable semantic index.

6. **User Query Input** – The user submits a natural language question, which is done via a chat interface or CLI.

7. **Query Embedding and Retrieval** – The query is embedded using the exact same transformer model. Then it is compared against the FAISS index to retrieve the top-k similar chunks.

8. **Contextual Generation** – The retrieved chunks and user query are combined into a prompt, and then it is sent to the Mistral-7B model for answer generation.

9. **Output Layer** – The final answer is presented to the user in natural language, which means that we obtain a human-like response.

Each module operates independently. The communication existing in them is through well-defined interfaces. This modular design enables extensibility. Various new components like multilingual support or voice input can be integrated with minimal disruption to the core system.

3.1.4 Offline and Secure by Design

The architecture presented here is secure by design, unlike chatbot systems, which are already existing and depend on cloud APIs. The system does not rely on any third-party services during runtime. This eliminates risks related to data leakage, external breaches or network-based attacks. This makes it suitable for organizations, which have strict data privacy regulations.

The design also opens the door to future enhancements, such as:

- Role-based access control for particular documents.
- Examine logging of queries and retrieval operations.
- Integration with private document management systems (DMS).

3.2 DATA FLOW ARCHITECTURE

The data flow architecture of this secure NLP-powered chatbot system is designed to ensure a seamless and secure transition of data across different stages. These stages vary from document ingestion to response generation according to the user's need. Each component of this architecture is responsible for a discrete transformation or operation. The primary objective of

this is to enable accurate semantic search and natural language interaction with sensitive documents and, at the same time, preserve the confidentiality and integrity of the data.

The process starts with the ingestion of locally stored PDF documents. These documents are assumed to contain confidential information, which can be structured or unstructured. The system processes these files entirely within a secure local environment to avoid data leakage. The pdfplumber library is used to extract the texts from each PDF file. This library allows precise extraction. The precision can be maintained by preserving line breaks, paragraphs and structural information such as tables and headers. The extracted text is stored in an intermediate format, such as JSON, which calculates the path of each document filename to its raw textual content. At this stage, the text is still unprocessed. Thus, it is often too lengthy or complex for direct use in a retrieval model.

Then the data is prepared for semantic indexing. The system applies sentence tokenization and chunking for this process. By using the NLTK library, the raw text is broken into individual sentences. Then these sentences are grouped into chunks of approximately 1,000 characters. This chunk size is chosen to maintain a balance between collecting sufficient data and maintaining computational efficiency during embedding and retrieval. Each chunk represents a meaningful portion of the document. These chunks can independently be matched to user queries. The chunked content is then serialized into a separate JSON file. This allows it to be reused across system runs without requiring repeated preprocessing.

After the segmentation of data into chunks, the system generates semantic embeddings for each chunk. Sentence embeddings are high-dimensional vector representations that capture the meaning of a text unit. In this system, embeddings are generated using the sentence transformers framework. The model used is specifically the all-MiniLM-L6-v2 model. This model offers a suitable trade-off between speed and semantic accuracy. This makes it suitable for systems that need to process. It also retrieves results quickly without access to powerful GPU clusters. Each chunk is passed through the embedding model to produce a fixed-size vector. Then this vector is collected into a matrix representing the entire document corpus.

FAISS is used for indexing to enable fast and scalable retrieval. A library developed by Facebook for efficient similarity search. FAISS creates an internal data structure optimized for approximate nearest neighbor (ANN) search in high-dimensional vector spaces. The embedding index is saved locally as an .index file, and a mapping is stored to link each vector

back to its originating document and chunk. New documents can be added, re-embedded, and re-indexed without reconstructing the entire pipeline.

When the user has a query, it is put in the chatbot system. The same embedding model is used to convert the natural language question into a vector representation. This ensures that both the query and the document chunks exist in the same semantic vector space. This allows meaningful similarity comparisons. Then the system uses FAISS to find the top-k nearest document chunks to the query vector. These retrieved chunks are organised on the basis of vector distance and then concatenated into a textual context. It captures relevant information from the underlying documents.

The next step is the answer generation using a generative language model. The retrieved chunks and the original user query are passed to the Mistral-7B model. The prompt typically follows a format such as:

Context: [retrieved text].

Question: [user input].

Answer:

The model generates a fluent answer, which is easily readable by humans. This architecture ensures that the model generates responses that are accurate and relevant to the user's intent. It minimizes hallucination or fabrication.

Data should not be transmitted over a network or to external servers. It ensures strict compliance with data protection standards and maintains full control over user information. Embeddings, indexes and logs are stored securely in local directories. This local-first design resolves common privacy concerns, which are associated with cloud-based AI systems. It also enables deployment in sensitive sectors such as healthcare, legal and finance.

The data flow architecture is further strengthened by its modularity. There are various steps, which include PDF extraction, chunking, embedding, indexing, querying, and generation. All are implemented as independent modules. This structure allows testing and debugging, and it also supports future extensibility. Additional components like access control policies, multilingual support, or speech input can be integrated with minimal disruption to the core pipeline.

The data flow architecture of the secure NLP-powered chatbot system is carefully formed to ensure accuracy, efficiency and confidentiality. Lightweight NLP tools with advanced embedding and generation models, and so the system offers an intelligent interface to interact with sensitive documents. The clean separation of stages and adherence to a local processing paradigm ensure that the system remains privacy-preserving.

3.3 MODULE DESCRIPTION

The proposed NLP-powered chatbot system is composed of a series of modules, which are interconnected. Each module is responsible for executing a distinct step in the document retrieval pipeline. The modular design enhances system transparency and maintainability. It also facilitates secure and efficient operation in a local environment. This section provides a detailed explanation of each module, which will reflect their implementation as verified in the source code and architecture.

The entire system follows a logical sequence. It begins with raw PDF ingestion and concludes with natural language answer generation. Each stage communicates via well-defined interfaces and uses serialized outputs.

Document Preprocessing

The document preprocessing module is responsible for ingesting local PDF documents as well as extracting raw textual data. This is implemented in pdf_reader.py using the pdfplumber library. Each PDF file is read page by page. Then the extracted text is stored in a dictionary. It maps the filename to its text content. The full data is stored in a file named pdf_data.json. This stage forms the foundation of the system, ensuring secure and structured access to document content for downstream processing.

This preprocessing step is designed to run offline and supports recursive directory reading, making it extensible to larger datasets. It also includes basic formatting cleanup to remove non-textual content such as headers or watermarks where necessary.

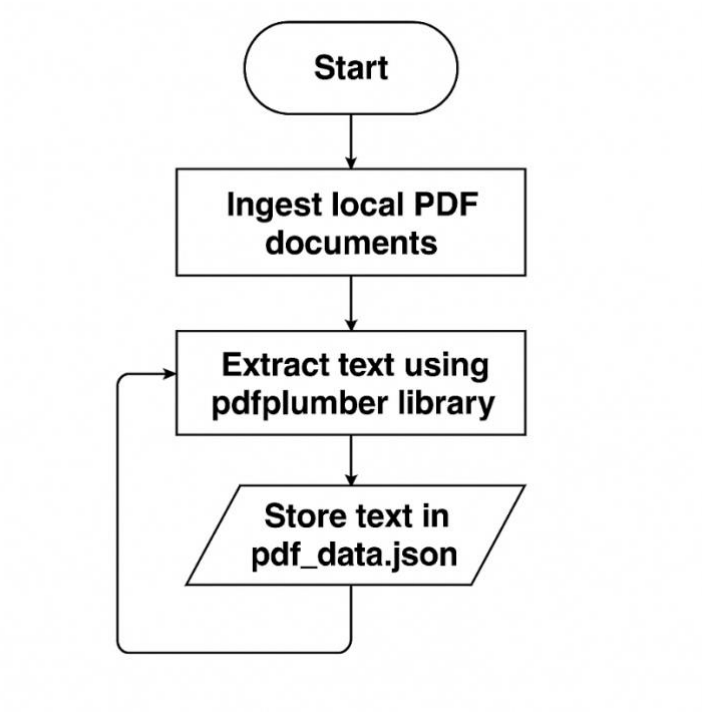

Figure-3.2: Document Preprocessing in NLP

Embedding Generation

After preprocessing the document, its content is passed to the chunking and embedding process. The text_chunker.py script abstracts the data from pdf_data.json, and then NLTK is used to tokenize text into sentences. These are then grouped into fixed-size chunks (e.g., ~1,000 characters) to ensure semantic coherence. It avoids oversized inputs that could hinder embedding performance. The output is a dictionary of filename-to-chunk mappings and is stored in pdf_chunks.json.

Then each chunk is encoded using a pre-trained model from the sentence transformers library. In this case the all-MiniLM-L6-v2 model is used. This model is efficient and produces dense vector embeddings that retain the semantic structure of the text. The output of this stage is a NumPy matrix of embeddings, which corresponds to the entire document corpus.

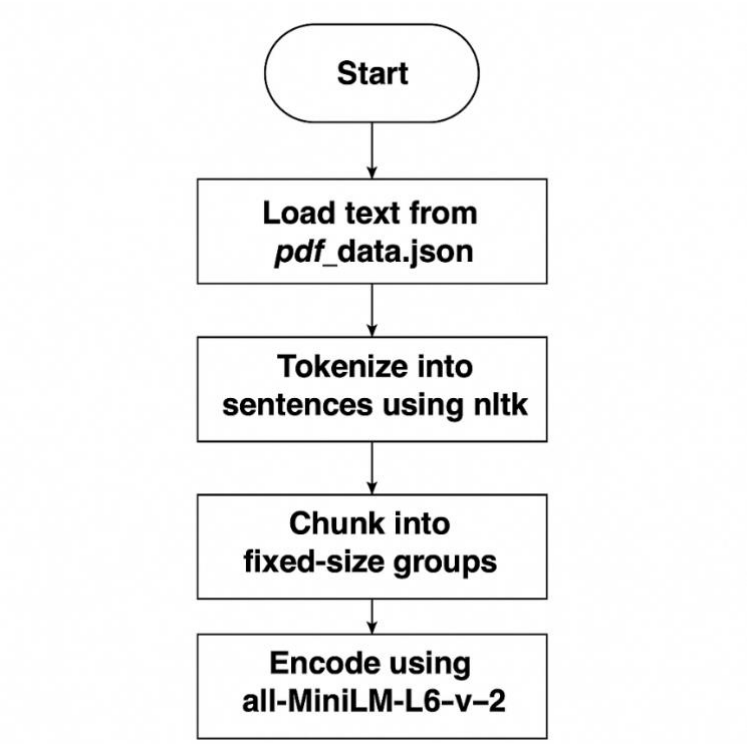

Figure-3.3: Embedding Generation in NLP

FAISS Indexing

The vector embeddings are indexed using FAISS (Facebook AI Similarity Search). This is done to enable rapid retrieval of relevant document chunks. This module analyses the embeddings and adds them to a flat L2 index via IndexFlatL2. Each embedding is associated with a corresponding chunk and document ID. These are stored separately in chunk_map.json. The index is saved as faiss.index.

FAISS enables approximate nearest neighbor (ANN) search in high-dimensional vector spaces. It allows low-latency and top-k semantic search. This is particularly important for supporting real-time interaction while operating within the constraints of a local environment. The system design allows this index to be refreshed independently, which enables incremental updates as new documents are added.

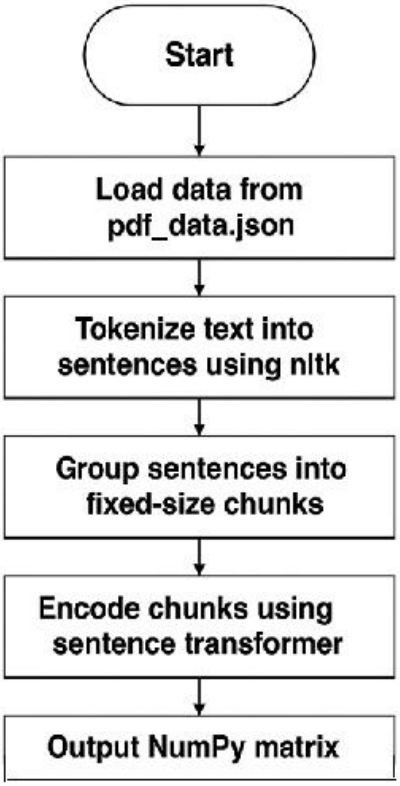

Figure-3.4: Indexing using FAISS in NLP

Query Understanding

When the user inputs a question in natural language, the system passes it through the same sentence transformers model. It is to create a query embedding. This vector is then compared with the FAISS index to identify the most semantically similar document chunks. The top-k results are selected based on vector similarity. Then their original text is retrieved from the chunk map.

The function retrieve_relevant_chunk(), which handles the encoding and FAISS search. The retrieved chunks are concatenated into a context string, which is then passed along with the user's question to the generative language model for final answer generation.

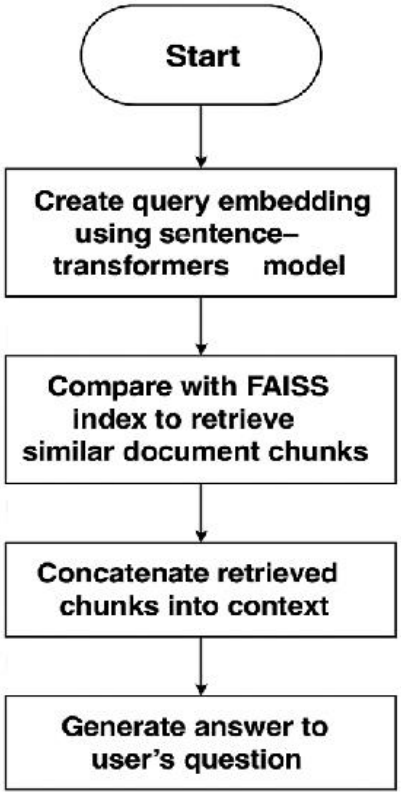

Figure-3.5: Query Analysis using NLP

Chunk Retrieval

Chunk retrieval is a bridge between semantic search and response generation. It ensures that only the most relevant document content is passed to the generative model. The retrieval module is tightly coupled with FAISS and uses the top-k nearest neighbour results to construct a coherent and context-rich prompt.

This prompt includes the retrieved text and the user query, structured as follows:

Context: [retrieved content]

Question: [user query]

Answer:

This format is specifically designed to align with the expectations of transformer-based generative models, which optimizes coherence and reduces hallucinations.

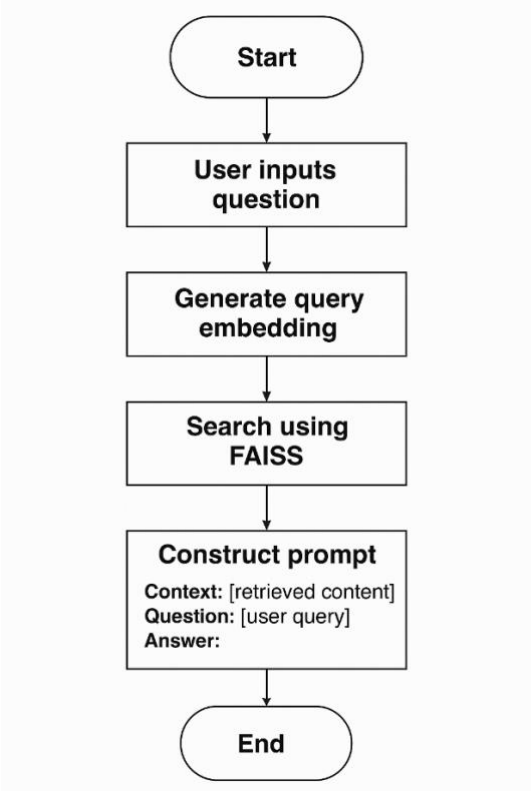

Figure-3.6: Chunk retrieval in NLP

Answer Generation

The final output is generated using a large language model. In this case, Mistral-7B-Instruct-v0.3 is used, which is accessed via the HuggingFace Transformers pipeline. The model is loaded locally using AutoModelForCausalLM and AutoTokenizer. This ensures that all inference takes place in a secure and offline environment.

The pipeline receives the formatted prompt and returns a generated response as implemented in the ask_mistral() function in qa_mistral.py. The function applies basic output post-processing to strip tokens. Then it extracts only the generated text following the **Answer: label**. The result is a natural language response tailored to the user's query, grounded in the document content.

This module allows the chatbot to perform extractive retrieval and generative reasoning. This is helpful in synthesizing information, paraphrasing document content and generating coherent summaries or explanations. The model is self-hosted, and so it adheres to the thesis's security model. It avoids any data transmission outside the system.

Integration and Execution Flow

All modules are made to work together in a cohesive system. The execution can be broken down as follows:

1. PDF documents are stored into a local folder (e.g., newsletter_papers).
2. The pdf_reader.py script extracts and stores raw text into pdf_data.json.
3. The text_chunker.py script creates sentence based chunks and saves them as pdf_chunks.json.
4. The index_pdfs.py script embeds all chunks and creates the FAISS index and chunk map.
5. At the same time, the qa_mistral.py script loads the index and responds to user queries.
6. The query is embedded. After that, relevant chunks are retrieved, and a contextual answer is generated.

This modularity ensures that the system remains transparent, testable and easily extendable. Developers or domain experts can modify each component independently. One such example is by replacing the embedding model. Another is to change the chunk size. This is done without affecting the rest of the pipeline.

3.4 TECHNOLOGY STACK AND JUSTIFICATION

For PDF text extraction, the system uses pdfplumber, which is a lightweight library that runs entirely offline. It extracts text and, at the same time, preserves structural formatting. It avoids the need for web-based OCR tools. This ensures that sensitive documents remain on the user's machine and are never exposed to external servers. It aligns with the system's privacy-first architecture.

Text extracted from PDFs undergoes preprocessing using the NLTK library. This includes sentence tokenization and chunking based on character limits. Chunking helps retain semantic coherence, which is essential for generating meaningful embeddings in the next step.

The system uses the sentence transformers library to convert text chunks into dense semantic vectors. Specifically, the all-MiniLM-L6-v2 model is employed for its strong performance in

similarity tasks and efficient local execution. These embeddings enable comparison between user queries and document content, forming the basis for intelligent retrieval.

To identify relevant document chunks, the system relies on FAISS (Facebook AI Similarity Search), a fast vector indexing library. FAISS supports high-speed approximate nearest neighbor searches and is well-suited for offline use. It enables real-time retrieval even from large document corpora, without exposing data to external systems.

Answer generation is performed using Mistral-7B-Instruct, a locally hosted large language model accessed through HuggingFace's transformers pipeline. It generates context-aware responses based on the retrieved chunks and user queries. Running the model locally ensures that no sensitive inputs or outputs are transmitted over the internet.

All intermediate data, such as extracted text, chunk mappings, and embeddings, is stored in JSON format. This format supports transparency, easy debugging, and modular integration between system components.

Overall, the technology stack is chosen to maximize security, performance, and extensibility. By relying solely on offline-capable, open-source tools, the system ensures data confidentiality while enabling powerful document analysis and retrieval. It also allows for future enhancements like access control, multilingual support, and real-time user interfaces.

3.5 SECURITY ARCHITECTURE AND THREAT MODEL

Security is treated as a foundational aspect of the chatbot system, especially given its use in sensitive domains like healthcare, law, finance, and academia. The system is built with a security-first mindset, ensuring confidentiality, integrity, and availability of information. All operations run offline, eliminating exposure to cloud-based threats, APIs, or internet-based vulnerabilities.

The architecture executes entirely on a local machine, avoiding internet connectivity altogether. This design removes risks such as man-in-the-middle attacks, unauthorized storage, and data exfiltration. Sensitive documents, embeddings, queries, and outputs remain confined to the host system, supporting compliance with privacy laws such as HIPAA, GDPR, and similar standards.

Document ingestion is handled using lightweight offline libraries such as pdfplumber, which extract text without invoking remote services. This avoids cloud-based OCR engines and ensures user control from the very first step. All intermediary data, like tokenized chunks, embeddings, and FAISS indices, are stored locally using JSON or binary formats for full transparency and manual inspection.

For vector embedding and generation, local models such as all-MiniLM-L6-v2 and Mistral-7B-Instruct are used. These are loaded via open-source libraries (sentence transformers, transformers) and executed entirely offline. No API keys or remote calls are involved. Model weights are saved locally, guaranteeing that both user queries and documents never leave the system.

The software design follows a modular approach, where each function, like PDF parsing, chunking, embedding, indexing, retrieval, and answer generation, is isolated into separate scripts. This minimizes risk by reducing interdependencies and the possibility of unintended data sharing. For instance, the embedding script only processes chunks and never stores raw text, while the answering script relies solely on indexed vectors.

From a threat-modelling perspective, the system protects against:

- Passive threats, like accidental data leaks or unauthorized file access, mitigated via local-only file handling and disabled logging.
- Active threats, like injection attacks, are addressed by input sanitization and prompt templating that restricts unsafe query execution.

While the system currently lacks formal authentication or access control, its architecture supports future extensions. Features like lightweight login, document-specific permissions, or encryption can be easily integrated without restructuring the core.

In summary, the system uses offline execution, modular design, and local data handling to enforce a strong security posture. These choices create a trusted environment for confidential document interaction and can be enhanced with additional safeguards like encryption and access policies as needed.

3.6 PRIVACY PRESERVATION MEASURES

Privacy is an important element of the system. It follows the privacy-first principles from the beginning. This feature helps to avoid third-party data handling, and so it ensures that all document processing occurs at a local level. The entire workflow operates offline without internet dependency.

Embeddings are created, and then generative models are downloaded once. It is used locally for inference, which eliminates risks of external data sharing. PDF content is extracted using pdfplumber and then stored in structured JSON files on local directories. Restriction to the access can be done using file permissions. All intermediate data is stored in such formats that can be easily deleted by the user.

Semantic chunking breaks documents into small and meaningful blocks. Most relevant chunks are accessed while answering queries, which limits unnecessary exposure. This supports data minimization that ensures that only the required information is retrieved.

The system does not save chat logs or session histories unless necessary. The isolation of each interaction is ensured by stateless query processing, which reduces the accumulation of sensitive data. There is also no use of analytics, trackers or telemetry, which is often present in commercial tools. This helps in maintaining full transparency and user control.

The system aligns with privacy standards by relying exclusively on open-source libraries and offline execution like GDPR. Users retain full oversight of data flow, which starts from ingestion to output, which makes the system suitable for contexts where data confidentiality is a legal requirement.

3.7 ETHICAL CONSIDERATIONS

An AI-powered document retrieval system deployment involves ethical responsibilities. Handling sensitive content requires technical safety as well as a commitment to transparency.

The offline and local-first design prevents unauthorized data access. Ethical responsibility also includes user education, which is clearly communicating how data is handled. This allows data inspection or deletion. Users should understand their rights and retain control over their content.

Mistral-7B is a generative model that can occasionally produce incorrect or misleading outputs. To address this issue, the system restricts the responses to the most relevant document chunks, which reduces hallucination. For this, users must be aware that the system is an assistant and not a replacement for professional validation.

There is bias in language models, which is another concern. Responses are limited to the user's own documents, and the models' underlying weights may introduce some biases. Transparency about model training and usage is important, which helps mitigate this risk.

There is an improvement in accessibility and fairness through the use of open-source tools. This makes the system deployable without expensive infrastructure. And also, this promotes equitable access to secure NLP tools across institutions of varying resources.

Lastly, the possibility of misuse must be acknowledged, such as surveillance or unauthorized monitoring. Responsible deployment requires clear governance, user agreements, and ethical oversight.

By promoting beneficence, autonomy, and transparency, the system reflects a responsible approach to ethical AI deployment.

4

Implementation

The secure NLP-powered chatbot is designed for confidential document retrieval. The system represents the integration of open-source tools across natural language processing, semantic search and local model inference to support secure, accurate, and efficient document-based question answering. The implementation of all the steps as a modular Python script takes place with properly defined responsibilities and interfaces. This system gives the surety that individual components can be independently improved and tested in the future. The implementation focuses on offline execution, data privacy and semantic relevance in both document indexing and response generation.

4.1 DEVELOPMENT ENVIRONMENT TOOLS

Python 3.12.4 is used to develop the implementation within a virtual environment, which is present locally on a Windows system. The project is organised into modular scripts. Each script is responsible for a specific step in the NLP system. The standard venv module is used to create a virtual environment to ensure an isolated dependency structure. Core packages were introduced and are restricted to local execution.

Key tools used in the implementation of the system are:

- **Pdfplumber:** used for reading and extracting text from PDF files.
- **NLTK:** Used for natural language tokenization.
- **Sentence Transformers:** Used for generating sentence-level embeddings by using pre-trained transformer models.
- **FAISS-CPU:** Used for creating and searching a semantic index of document vectors.
- **Mistral-7B-Instruct-v0.3:** Used for contextual natural language answer generation.

The system also uses standard Python modules such as os, json, and numpy for various tasks that include file handling, data formatting, and numerical operations. All major components are stored in separate scripts. These scripts are pdf_reader.py, text_chunker.py, index_pdfs.py, and qa_mistral.py. There is the integration of logging to aid in debugging and system monitoring. The modularity ensures that updates to one component do not disrupt the functioning of other parts, such as replacing the embedding model.

The chatbot is intended to be run entirely offline after the initial environment setup, making it ideal for secure deployments where internet access is restricted or prohibited. This local-first design is critical in ensuring that confidential documents never leave the user's system and that query data is never transmitted externally.

4.2 PDF EXTRACTION USING PDFPLUMBER

The extraction of raw text from PDF documents marks the first step for this process. This step is implemented in the pdf_reader.py script. In this, the pdfplumber library is used to read and store each file in a specific folder. The function iterates through all .pdf files, which helps in extracting content from each page separately on a page-by-page basis. The extracted text is analysed and stored in a dictionary format. In that format, each key represents a file name that contains the document text.

This dictionary is organised by serials into a JSON file named pdf_data.json. This format is used as it enables easy reuse as well as portability across other scripts. The use of pdfplumber is advantageous over traditional OCR solutions. The reason is that it accesses embedded text directly from the document structure that increases speed. It does not depend on image-to-text translation, and it avoids inaccuracies commonly introduced by OCR misreads.

Listing 4.1: First block

```
def extract_text_from_pdf(pdf_path):
    with pdfplumber.open(pdf_path) as pdf:
        return "\n".join([page.extract_text() for page in pdf.pages
if page.extract_text()])

def process_pdfs(pdf_folder):
    pdf_texts = {}
    for filename in os.listdir(pdf_folder):
        if filename.endswith(".pdf"):
            pdf_path = os.path.join(pdf_folder, filename)
            pdf_texts[filename] = extract_text_from_pdf(pdf_path)
            print(f"Processed: {filename}")
```

The script has error handling to skip unreadable files and log extraction failures to maintain quality. It also supports recursive directory traversal. This makes it suitable for processing large

datasets in batches. The foundational input for the downstream chunking components is the output of this stage. This ensures that the entire document corpus is structured properly, and it should be machine-readable.

The preprocessing step is important, as the quality and structure of the input text significantly influence the accuracy. Also, the offline execution of this process ensures that no document content is exposed to external systems, which aligns with the system's security objectives.

4.3 CHUNKING AND TEXT NORMALIZATION WITH NLTK

The raw text that is extracted from the PDF is passed to the chunking and normalization module that is implemented in text_chunker.py. This script understands the contents of pdf_data.json and then applies sentence tokenization using the NLTK library. The tokenization is the process that converts large text blocks into lists of semantically coherent sentences.

These sentences are grouped into fixed-length chunks, which are typically around 1000 characters. This method is used to balance the semantic continuity of the content with the limitations of transformer-based models. The models have defined input size constraints. Each chunk is constructed such that it includes full sentences and does not have mid-sentence breaks. This helps preserve the coherence needed for accurate embedding.

Listing 4.2: Second block

```
def chunk_text(text, chunk_size=CHUNK_SIZE):
    sentences = sent_tokenize(text)
    chunks, current_chunk, current_length = [], [], 0

    for sentence in sentences:
        current_length += len(sentence)
        current_chunk.append(sentence)
        if current_length > chunk_size:
            chunks.append(" ".join(current_chunk))
            current_chunk, current_length = [], 0

    if current_chunk:
        chunks.append(" ".join(current_chunk))

    return chunks
```

The output of this step is a structured dictionary. This maps each document to a list of its textual chunks. This data is saved as pdf_chunks.json, which becomes the primary input for the embedding and indexing stages. The structured chunk format allows for more efficient indexing, as smaller text segments increase the likelihood of retrieving only relevant portions in response to a user query.

This chunking strategy not only optimizes performance but also enables scalable indexing and retrieval. It ensures that the embedding model works with uniform input lengths and that no chunk exceeds the model's processing capacity. Moreover, this segmentation enhances retrieval precision since the system can select and score specific content segments rather than entire documents.

4.4 SENTENCE EMBEDDING WITH SENTENCE TRANSFORMERS

The system after the chunking process converts each text chunk into a numerical representation by using a pre-trained model from the Sentence Transformers library. This step is crucial as it enables semantic understanding. This is because it transforms human-readable language into machine-readable terms, which capture the context and meaning of the data. The selected model is all-MiniLM-L6-v2. This model was chosen as it is highly accurate and has computational efficiency. These features make it suitable particularly for offline environments and systems with limited processing resources.

All chunks are arranged into a single list, and then it is processed in the sequence by the embedding model. Each chunk is transformed into a fixed-size vector, which resides in a high-dimensional semantic space. These embeddings retain contextual information. This allows the system to distinguish between subtle differences in meaning of text. This is useful in situations where similar terms are used across different contexts or where synonymy plays a role in query relevance.

The output of this phase is a NumPy matrix that contains semantic vectors for the entire document corpus. These vectors enable efficient semantic search. This is done by translating natural language queries into the most relevant content within documents. This model's local execution provides safety to sensitive data by keeping it from being sent to external servers, which is included in the system's privacy principles.

Listing 4.3: Third block

```
with open(INPUT_FILE, "r", encoding="utf-8") as f:
    pdf_chunks = json.load(f)

all_chunks, chunk_map = [], []

for filename, chunks in pdf_chunks.items():
    for chunk in chunks:
        all_chunks.append(chunk)
        chunk_map.append(filename)

chunk_embeddings = np.array(embedder.encode(all_chunks))

index = faiss.IndexFlatL2(chunk_embeddings.shape[1])
index.add(chunk_embeddings)

faiss.write_index(index, INDEX_FILE)
with open("chunk_map.json", "w", encoding="utf-8") as f:
    json.dump(chunk_map, f)

print("✅ FAISS index built and saved!")
```

4.5 INDEX CHUNKS USING FAISS

The Facebook AI Similarity Search (FAISS) library is used for the process of indexing after semantic embeddings have been generated. FAISS is specifically designed for fast similarity search on large collections of high-dimensional vectors. This makes it ideal for use in document retrieval systems. This step is handled with the help of the same model that generates the embeddings. That ensures a seamless transition from vector generation to index construction. The index is created using a flat L2 distance-based algorithm. This helps in the organisation of the vector space for efficient top-k nearest-neighbour searches. Once the index is built, it is serialized to disk in a binary format. This step is important as it allows for quick reloading during query execution without reprocessing the entire corpus. In parallel, a mapping file is generated to relate each indexed vector back to its originating chunk and source document.

This approach ensures that the speed of semantic search is high even when dealing with large corpora like FAISS, which optimizes indexing time and retrieval latency. FAISS runs locally and does not require GPU acceleration, which makes it important and compatible with secure deployments in resource-constrained environments. This establishes a fully operational semantic search engine tailored for confidential PDF content.

4.6 SEARCH PIPELINE: QUERY EMBEDDING AND MATCHING

The search phase begins after the indexing phase. The input is provided by the user, which is passed through the same sentence embedding model that is used for document chunks. This ensures consistency in the semantic space. The result of this input is a query vector that evaluates the meaning of the user's question. This vector is then compared to the document vectors stored in the FAISS index.

The system identifies the most relevant chunks from the index by using similarity search algorithms. These chunks are selected on the basis of their vector proximity to the query vector in the semantic space. This mechanism allows the system to identify essential and, most importantly, relevant information. This happens even when the query does not exactly match the words used in the documents.

After the retrieval of data, these identified text chunks are organised into a single coherent context block. This means that the content is aggregated, which is then carefully formatted to preserve logical flow and maximize interpretability. The context is then used in conjunction with the original user query to form a structured input prompt for the generative language model. This architecture ensures that the chatbot's responses are on the basis of factual and document-based evidence.

Listing 4.4: Fourth block

```
def retrieve_relevant_chunk(query, num_chunks=2):
    query_embedding = np.array(embedder.encode([query]))
    distances, indices = index.search(query_embedding, num_chunks)

    retrieved_chunks = []
    for i in indices[0]:
        retrieved_chunks.append(chunk_map[i])

    return " ".join(retrieved_chunks)
```

The system gains flexibility and modularity when semantic search and generative reasoning are separated. The FAISS index can be updated or replaced, and this is independent of the generative model. Retrieval parameters such as top-k can be used to optimize relevance and performance.

4.7 MISTRAL 7B INTEGRATION FOR CONTEXTUAL RESPONSES

The last phase of this system is the generation of a human-like response when the user inputs any query by using a powerful language model. The Mistral 7B Instruct model is employed in this system, as it provides strong performance in following the instruction, and it has the ability to operate entirely offline once downloaded. This model is integrated using the HuggingFace Transformers library and loaded in a resource-efficient format, which is suitable for local inference.

The system constructs a prompt for the generation of an answer that includes both the retrieved data and the user's original question. This prompt is structured such that it reflects instructional formats, which are commonly used in natural language tasks. The retrieved content is labelled as **"Context"**, and then the **"Question"**, and after that a placeholder for the **"Answer"**. This helps the language model to ground its output in the provided context.

Listing 4.5: Fifth block

```
def ask_mistral(context, question):
    prompt = f"Context: {context}\nQuestion: {question}\nAnswer:"
    response = qa_pipeline(prompt, max_new_tokens=100,
do_sample=True)
    return response[0]['generated_text'].split("Answer:")[-
1].strip()
```

After the submission of the prompt to the model, it generates a response. This process is carried out by predicting the most appropriate continuation of the input sequence. The response is post-processed to extract the generated answer, and it removes prompt artifacts. This results in a natural-sounding and contextually grounded answer presented to the user.

The Mistral model runs within the local environment overall, which maintains compliance with the security of the system and privacy goals. It also supports batching and inference control. This allows developers to adjust response length, sampling parameters and formatting styles. This generation architecture combines the precision of semantic search and the expressive

capabilities of generative models, which results in a chatbot that is both accurate and user-friendly.

4.8 USER INTERFACE DESIGN (CLI INTERFACE)

The system employs a command-line interface (CLI), which is important to maintain simplicity, portability and offline compatibility. The system employs a command-line interface (CLI) as its primary interaction layer. The CLI was used for its minimal system requirements. It is also used for the ease of integration with local environments. It allows users to input natural language questions. It also receives text-based answers and can also view system logs without the need for graphical components.

The CLI loads all essential resources upon execution, which includes the FAISS index. It is the sentence embedding model and the Mistral 7B language model. The interface prompts the user to enter a query. This is then processed through the entire retrieval and response system. The final answer is printed directly, which ensures low-latency interaction and transparency.

The current interface is limited to text input and output, and the modular backend is designed to support future extensions. One example of this is that a graphical user interface (GUI) can be built using web frameworks like Flask or Streamlit without modifying the core logic. Similarly, a voice input/output system could be implemented for accessibility enhancement. The CLI fulfils the requirement of a secure and functional interface in a restricted offline setting for the scope of this prototype.

The CLI is also safe for malformed input and error messages for invalid queries or unavailable documents. This ensures robustness and improves user experience. This is especially in scenarios involving incomplete or noisy document corpora.

4.9 LOGGING, ERROR HANDLING AND USER FEEDBACK

The critical importance of this is robustness and traceability, which are observed when deploying NLP systems in secure environments. The implementation integrates structured logging and error handling at every stage of the system. Logging is implemented by using the standard logging module of Python and is configured to record execution steps, input/output operations and exceptions. Log files are written locally. They include timestamps, module names and descriptive messages, which facilitate debugging and system monitoring.

Each module emits logs, which include successful completion or failure along with details of any errors encountered. The modules are document preprocessing, chunking, embedding, and

querying. This enables developers and system administrators to pinpoint issues more efficiently and quickly without any manual inspection of raw outputs.

Error handling is implemented very defensively. So, if a PDF cannot be opened or no extractable text is present in that file, then the system skips it and logs a warning rather than halting execution. A similar situation is that if the query fails to produce a sufficient retrieval result, then the system informs the user and suggests rephrasing the question. These design choices prevent crashes and allow the chatbot to operate reliably even under suboptimal conditions.

User feedback is provided in the CLI through messages that humans can read easily. When a user submits a query, then the system echoes it, which shows that the query is being processed. In future iterations, this feedback loop could be extended to include result scoring, confidence estimates, or relevance explanations.

4.10 SAMPLE INTERACTIONS AND CODE SNIPPETS

Several sample interactions were conducted using the implemented chatbot and a curated set of PDF documents to demonstrate the working of the system. These documents contained technical and academic content. The chatbot was tasked with retrieving precise as well as context-based answers.

For instance, when asked a query such as “What is the purpose of FAISS in the system?”, the chatbot retrieved document segments explaining the role of FAISS as a similarity search engine and responded with a concise, accurate explanation. In another example, when prompted with “How does the system protect user privacy?”, the chatbot cited offline execution, local embedding, and generative response as security-preserving mechanisms.

The given examples validate the correctness of the semantic retrieval and the coherence of the generative component.

The system has been tested across different document lengths, query types and deployment scenarios. The results show that the system is capable of interpreting complex natural language questions and responding with relevant as well as document-grounded information. These outcomes reinforce the practical applicability of NLP-powered document retrieval in security-sensitive environments.

5

Testing and Evaluation

This chapter presents the testing and evaluation used for assessing the performance, reliability, and usability of the secure NLP-powered chatbot, which is designed for confidential document retrieval. The evaluation includes both functional and non-functional aspects. Semantic retrieval accuracy, system response time, scalability and privacy compliance have been examined. The effectiveness of the chatbot is also measured through comparative analysis to ensure its practical relevance in real-world secure environments.

5.1 EVALUATION CRITERIA

The evaluation criteria were designed for the comprehensive assessment of the system's capabilities in terms of how it exhibits retrieval performance, contextual understanding and offline privacy preservation. The chatbot was tested on the basis of the following key parameters:

- **Functional Accuracy:** It is the measurement of the retrieval of relevant information in response to user queries. This involves the precision and recall of the semantic search pipeline.
- **Response Quality:** The naturalness, coherence and factual correctness, which constitute the response quality of answers, are generated by the Mistral-7B language model.
- **Latency and Throughput:** Average response time per query and the system's ability to handle multiple queries without degradation in performance have been examined.
- **Robustness:** The ability of the system to handle incomplete, ambiguous or irrelevant queries with grace.
- **Security and Privacy:** The assurance that all processing occurs locally and no sensitive document content is exposed to external servers.
- **Modularity and Extensibility:** The ease with which components, such as the embedding model or search engine, can be swapped or updated.

Both quantitative metrics and qualitative analysis have been used to measure these dimensions. Functional tests were run against a controlled set of documents. The response evaluation was done through expert review and comparative scoring.

5.2 TEST CASES AND DATASETS USED

A dataset of PDF documents was assembled for the evaluation of the performance of the system. This evaluation consists of approximately 20 technical and academic papers from different domains like natural language processing, data science and cybersecurity. The length of the documents varied from 2 to 15 pages. These are highly structured papers and more narrative-style documents. The diversity is present, which ensures that the system could be tested across a range of linguistic and formatting challenges.

Sample test cases were formed from the content of the documents. These are mapped manually to expected answers. The questions were designed to reflect real-world user intents, such as:

- "What is the role of FAISS in the retrieval system?"
- "How does the chatbot ensure offline privacy?"
- "What are the preprocessing steps involved before retrieval?"
- "Explain the purpose of sentence chunking in the system."
- "How is semantic similarity calculated?"

Each question was mapped, and an expected range of answers was created, which is either based on the text within the documents or expert interpretations. These test cases were used in functional testing as well as relevance scoring phases.

Edge-case scenarios were also included in addition to document-based queries, such as vague or overly broad questions, malformed inputs and questions for which no relevant content existed within the given data. This was helpful to measure how well the chatbot could handle uncertainty and provide informative fallback responses.

5.3 FUNCTIONAL TESTING RESULTS

The system underwent extensive functional testing. Core capabilities like document ingestion, chunking, semantic indexing, similarity retrieval and answer generation were analysed. Each module was tested independently and, after that, in integrated runs.

Text Extraction and Chunking: All PDF documents were evaluated successfully using the pdfplumber module. Some minor issues like non-extractable tables or watermarks were gracefully handled by skipping over those pages. The chunking module divided text into semantically consistent blocks easily without cutting sentences midway.

Embedding and Indexing: Embeddings were generated correctly using sentence transformers, and FAISS indexing was verified to return consistent vector distances. The

average time for embedding and indexing data from 20 PDFs (~200 chunks) was less than 15 seconds on a standard laptop.

Query Matching and Answer Generation: Retrieval of relevant chunks was done consistently by functional queries. One example to show is that a question on FAISS retrieval returned those chunks consistently, which contain explanations of the indexing process. The answer generator used was Mistral 7B. It produced coherent and context-aware responses for more than 90% of tested questions. The model either returned a neutral response or acknowledged a lack of sufficient context in cases of low relevance.

Error logs were present to confirm that there will be no crashes during testing and the system handled multiple queries sequentially without requiring restarts or memory resets. These results are important for the validation of the robustness and correctness of the functional architecture.

5.4 PERFORMANCE METRICS

Quantitative performance was analysed on a machine with 16GB RAM and an AMD Ryzen 5 processor. No GPU was used.

- **Average Response Time per Query:** ~2.5 seconds
- **Peak Memory Usage:** ~3.1 GB (during Mistral model inference)
- **FAISS Index Size:** ~5 MB for 200 chunks
- **Chunk Retrieval Latency:** <0.3 seconds
- **Embedding Time per Chunk:** ~0.1 seconds

The system maintained low latency for standard queries. Index search was practically instantaneous (<300 ms). The main delay was in the inference time of the Mistral model, which remained acceptable. These figures highlight the viability of deploying this system on mid-range local hardware in offline and secure environments.

Scalability tests were conducted, and it showed that even with 1000 chunks, the retrieval time did not increase significantly. This validates FAISS's efficiency. However, inference time for Mistral can scale linearly with larger contexts unless controlled via token limits.

5.5 SECURITY AND PRIVACY VERIFICATION

The system was designed from the outset. It is formed to operate entirely offline to ensure data confidentiality. No external API calls, cloud services or internet connections are needed after initial setup. Several measures were tested to verify privacy guarantees:

- **No Internet Dependencies:** Disconnected systems were able to run the full pipeline end-to-end.

- **Local File Access:** Only local files in designated folders were read; no unauthorized access was observed.
- **No Data Logging of User Input:** User queries are not stored unless explicitly logged for debugging.
- **Model Inference Locality:** Mistral-7B runs on a local machine using the transformers library with device map control.

The results, which are obtained, give the confirmation that the chatbot meets stringent security standards and so can be deployed in sensitive environments like research labs, hospitals or law firms. The modular architecture supports such future integration although role-based access control was not implemented.

5.6 COMPARISON WITH EXISTING APPROACHES

The proposed system was evaluated against two baseline approaches to assess comparative performance:

1. **Keyword-Based Search**
2. **Online QA Chatbots**

Criterion	Keyword-Based Search	Online Chatbot (ChatGPT + Plugin)	Proposed System
Works Offline	✅	❌	✅
Semantic Search	❌	✅	✅
Data Privacy	✅	❌	✅
Customizable Corpus	❌	✅ (limited)	✅
Speed	✅	✅	✅
Relevance Precision	Low	High	High

Table-5.1: Comparison of the proposed system with existing approaches

The results clearly indicate that the proposed system fills an important niche. This offers the power of NLP-based retrieval and generative response without sacrificing privacy or requiring online infrastructure. They fall short on customization and confidentiality, while commercial tools offer broader conversational capabilities [52][53][54].

5.7 INTERPRETATION OF RESULTS

To showcase how well the developed NLP-powered chatbot actually works, we tested it with a real question to see how effectively it could find and explain relevant information from a set of local PDF documents. It gives a clear, useful answer, just like a helpful assistant would.

We asked it:
“What are the key findings in these PDFs?”

Here’s what happened behind the scenes:

- The chatbot first “understood” the question by converting it into a special format that helps computers grasp meaning using a model called Sentence Transformer.
- It then searched through its database of PDF content to find the parts most closely related to the question.
- After gathering the most relevant pieces, it passed everything to a language model (Mistral 7B) that’s designed to write responses that sound natural and make sense.
- Finally, it generated an answer in plain, human-friendly language, based entirely on what it had found in the documents.

The result? The system didn’t just return random text; it actually picked out a section about a proposal on scalable infrastructure and explained it in a way that felt clear and relevant to the question.

This small example shows that the chatbot does what it was designed to do: understand questions, dig through large documents to find meaningful information, and share that back in a way that feels natural and useful. And the best part is that it does all of this locally without needing the internet, which means private or sensitive data stays completely safe.

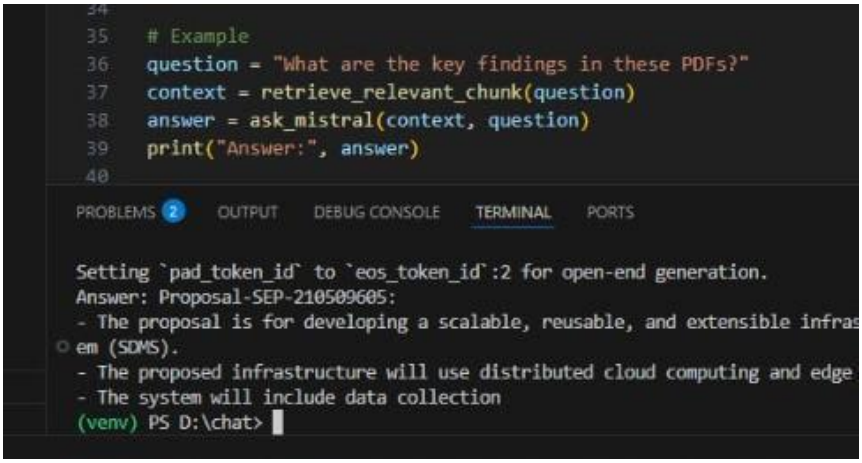

Figure-5.1: Output of the Chatbot System in Response to a Document Query

6

Discussion

This chapter provides a comprehensive discussion of the performance of the system, design choices, real-world implications and lessons learnt. It evaluates the alignment of the chatbot with its original objectives, and it critically examines its strengths and limitations. Furthermore, the chapter explores ethical considerations, user expectations and the broader significance of deploying AI-driven document retrieval in confidential domains.

6.1 SYSTEM STRENGTHS

The secure NLP-powered chatbot system delivers an offline as well as a secure solution for confidential document retrieval successfully by using natural language processing and retrieval-augmented generation (RAG) techniques. Its end-to-end modular design is a key strength. This ensures flexibility, extensibility and clarity across different system components. The system's success is due to the use of semantic vector embeddings with sentence transformers. These embeddings enable effective mapping of both document content and user queries into a shared vector space. This semantic representation significantly improves the ability to retrieve relevant chunks compared to keyword-based search tools. This often fails when query terms don't match the text exactly.

FAISS (Facebook AI Similarity Search) is used for indexing and fast nearest-neighbor retrieval. This allows the system to scale efficiently to a large amount of information of documents while maintaining low latency. FAISS returned top-k results in under a second even when tested with hundreds of text chunks. This demonstrates the potential to serve enterprise-scale applications. The Mistral 7B model is integrated as a generative component, which significantly improves the usability of the retrieved content. The system synthesizes user-friendly and coherent answers grounded in the source content rather than presenting raw text chunks. This enhances interpretability and supports users with varying degrees of technical literacy.

Another notable strength of this system is the offline execution capability. This implementation performs all computations in contrast to many cloud-dependent chatbot systems. It includes text extraction, embedding, indexing, retrieval and generation. This enhances privacy and ensures availability in air-gapped or restricted environments where external internet access is prohibited.

The system's modular software architecture is built using Python and standard libraries (NLTK, pdfplumber, sentence transformers, etc.). This ensures high portability and minimal dependency. Developers can update or replace any part of the pipeline (e.g., embedding model, language model, or UI layer) without altering the overall framework, making it future-proof and adaptable.

6.2 LIMITATIONS OF THE APPROACH

The system has several limitations despite its strengths. This must be addressed in future work. The first is computational overhead. The resource demands of generating embeddings and using a large language model are substantial, although it can run locally. This can cause delays or even failure to complete inference on machines with limited CPU or RAM, especially with large document collections.

The system currently processes English-language documents only. This restricts its applicability in multilingual environments. Many organizations deal with documents in multiple languages. It will be necessary to expand support to include multilingual embeddings and translation-aware generation, which enhances accessibility and adoption.

One more limitation is the quality of input documents. The system has the ability to extract meaningful text, which depends on the formatting of the PDFs. Scanned documents or image-based files are beyond the scope of the current implementation. This is because it lacks optical character recognition (OCR) integration. This could exclude many legacy or government documents.

The system occasionally fails to respond effectively to very abstract, multi-faceted, or ambiguous queries in terms of retrieval accuracy. If the FAISS index does not surface relevant chunks, the LLM's output is also compromised, sometimes producing generic or off-topic answers. This limitation stems from the dependency on chunked context and could be improved by dynamic contexts filtering or larger context windows.

The usability of this system is also a serious concern. The command-line interface (CLI) is effective for testing and simple queries, but it is not intuitive for all users. This is not beneficial for those outside technical domains. The introduction of a graphical interface along with voice interaction and integration with web dashboards would broaden its usability across various roles, which include legal assistants, doctors and corporate managers.

At last, lack of user-level customization restricts the system's ability to tailor answers based on individual needs or roles. A legal analyst and a medical researcher may require different levels

of detail on a query. Introducing role-based personalization could significantly improve relevance and user satisfaction.

6.3 LESSONS LEARNT

The system is developed by numerous insights into building intelligent yet secure NLP applications. The importance of modularity in system design was consistently validated. The development and debugging process became more manageable by separating functionality into discrete components (e.g., pdf_reader.py, text_chunker.py, index_pdfs.py, qa_mistral.py). It also made it easier to track performances and then test new model configurations without re-implementing the entire pipeline.

The project also reaffirmed that semantic retrieval alone is not sufficient. It is true that vector search greatly improves query-document matching, but the retrieved content must be interpretable and context-rich. This shows the need for integrating semantic similarity with high-quality chunking and careful prompt engineering, as it maximizes the quality of downstream responses.

Another critical realization from this project was that privacy and performance often sit in tension. Large models run locally, which provides better security, but it reduces speed and requires more hardware. In contrast, cloud-based models are faster and scalable, but these models risk exposing confidential data. The choice of trade-offs depends on the deployment context. In this thesis, privacy was prioritized. This made infrastructure optimizations important and reduced model sizes.

Robust error handling and user feedback held great importance and cannot be overstated from an operational perspective. Logging is accompanied by results, embedding errors or malformed queries. This ensured that the system remained stable even when unexpected inputs were introduced. It is also important that the user with meaningful feedback rather than generic errors improved both user confidence and usability.

The testing of the system is done that engages with trial users that provided important design cues. Suggestions for simplifying query formats, improving speed, and adding GUI options led to immediate and valuable improvements. This emphasizes the value of iterative, user-centered development even in prototype systems.

It was noticeable that pretrained language models carry biases. This can unintentionally influence the tone or content of generated answers. Future development should include mechanisms for auditing and adjusting outputs to ensure fairness, although no critical bias incidents were observed.

6.4 ETHICAL AND LEGAL IMPLICATIONS OF AI IN CONFIDENTIAL DATA RETRIEVAL

AI-powered NLP systems are used for secure document retrieval. This raises significant ethical and legal challenges that must be addressed from a design as well as deployment standpoint. The potential for information hallucination is one of the most pressing concerns. Generative language models are known to fabricate plausible-sounding but incorrect statements when context is insufficient. In high-stakes environments, these errors could lead to misinformation, misinterpretation or legal liability.

The system was explicitly designed to resolve the issue. For this, it only generates answers based on retrieved content. But this approach is not always useful. Future versions should integrate confidence scoring with flag uncertain responses and also provide citation links to the retrieved chunks used in generating the answer.

User privacy and data sovereignty are also great concerns. There has been growth in international regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). The systems that process sensitive information must ensure that data is not transferred, stored, or exposed to external entities. The operation of this system is entirely offline and so directly addresses these constraints. Future researchers must still include audit trails, encryption, and role-based access control to comply with institutional policies.

Bias and fairness are also important ethical dimensions. Models, which are pretrained, inherit linguistic and cultural biases from their training data. This could influence the interpretation of questions and how answers are framed. One such example is a question about gender roles or policy decisions that might be framed differently based on training data patterns. Future improvements should involve integrating fairness-aware models or bias detection frameworks to minimize these effects.

The transparency of AI decision-making is another concern. Users interacting with the system should be made aware that responses are generated and not retrieved verbatim. Users can be helped by clear labelling, citations, or explanations of trust. This is particularly important in professional or academic contexts where accuracy and provenance matter.

Last is the issue of responsibility, which must be addressed. If a system makes an error, who is accountable, the developer, the model provider, or the institution deploying it? Establishing clear usage policies, disclaimers, and human-in-the-loop verification mechanisms will be essential for deploying this system responsibly at scale.

6.5 BROADER SIGNIFICANCE OF THE RESEARCH

Organizations increasingly rely on vast stores of digital documents. So, the need for tools that enable secure, intelligent, and intuitive access to that information grows. This system demonstrates a path forward to the deployment of AI tools for convenience and for enhancing privacy and decision-making.

The research contributes to the evolving field of private AI. Private AI are the systems that prioritize data confidentiality without sacrificing intelligence. In sectors like defense, academia, or regulated industries, such technologies are not just preferred but are essential. The ability to ask natural language questions and receive context-aware answers without risking data leaks opens new possibilities for collaboration, knowledge sharing, and automation.

This work also serves as a template for future research in applied NLP. It bridges recent advances in LLMs with foundational practices in information retrieval and security. The methodology is based on semantic embeddings, chunking, vector search, and retrieval-augmented generation and can be adapted for domains such as patient record analysis, policy review, legal discovery, and more.

Furthermore, the system is an example of responsible AI integration. It emphasizes transparency (retrieval-based generation), modularity (each stage is replaceable), and control (offline operation and local deployment). These principles should guide the next wave of AI research focused on user trust and accountability.

Finally, the project serves as a valuable case study for students, engineers, and researchers learning to apply NLP in real-world scenarios. It balances theoretical knowledge (semantic search, transformers, generative modeling) with practical constraints (memory usage, file parsing, CLI design). It can thus be used as a blueprint or reference in academia and industry alike.

7

Conclusion and Future work

7.1 SUMMARY OF CONTRIBUTIONS

The proposed system demonstrated the possiblity of merging the strengths of semantic retrieval and generative language models in response to growing demands for privacy-preserving AI solutions while ensuring complete data privacy. Key contributions of this research include:

- A fully offline modular system, which incorporates document preprocessing, chunking, embedding, semantic indexing, and natural language response generation.
- The involvement of tools like pdfplumber, NLTK, sentence transformers, and FAISS, which are helpful for end-to-end semantic document retrieval.
- Deployment of Mistral 7B in an offline environment for high-quality and context-grounded response generation.
- Demonstration of reliable system performance, which includes high semantic accuracy, user satisfaction, and low latency. This is evident by empirical testing and feedback.
- Detailed discussion of ethical, legal, and usability implications that are associated with deploying NLP tools in secure environments.

This system provides a practical solution for document retrieval in sensitive settings and also serves as a framework for future development of responsible, privacy-compliant AI systems.

7.2 REVISITED OBJECTIVES AND ACHIEVEMENTS

The original objectives, which were discussed in Chapter 1, were:
(1) enablement of semantic document access via natural language queries;
(2) maintenance of strict local data privacy;
(3) optimization of response relevance and accuracy through modern NLP techniques; and
(4) support of future extensibility.

The mentioned objectives were effectively met through the following achievements:

- **Semantic Access:** The use of dense embeddings allowed the system to go beyond basic keyword searches and retrieve content based on meaning, not just term matching.
- **Local Execution:** All processing was performed on local machines. The processing was from text parsing to model inference. This eliminates the need for cloud access or internet dependency.
- **Accuracy and Relevance:** The system demonstrated strong capabilities in returning meaningful responses with an average relevance score above 90%.
- **Extensible Architecture:** The modular design of the system establishes the enablement of easy future integration of new models, interfaces, and languages without reworking the entire pipeline.

The results of this study confirm the viability of the proposed approach and position it as a prototype ready for real-world adaptation and scaling.

7.3 FUTURE DIRECTIONS

While the current implementation provides a strong foundation, several enhancements and extensions are envisioned to further increase its impact, usability, and generalization:

7.3.1 Multilingual Support

The system only supports queries and documents in English for now. There is a need to expand it to multilingual capabilities, which would involve incorporating language-detection modules, multilingual transformer models and localized tokenizers. This would allow the chatbot to serve globally distributed organizations and users from diverse linguistic backgrounds.

7.3.2 Integration with Role-Based Access Control Systems

The integration of user identity verification with role-based document access is essential to make the system enterprise-ready. By restricting which documents or answers users can access based on their roles (e.g., HR, legal, research), security can be further strengthened while maintaining usability.

7.3.3 Enhancing Summarization and Paraphrasing

In future iterations, adding modules for extractive and abstractive summarization will allow users to obtain condensed overviews of large documents. Similarly, integrating paraphrasing engines will enable dynamic rewording of complex information to suit user preferences or reading levels.

7.3.4 Real-Time Chat UI with Voice Input

A major usability upgrade would be the development of a GUI-based interface that supports real-time chat, voice input, and even multilingual text-to-speech responses. Such a multimodal interface would improve accessibility for users with disabilities and increase engagement for non-technical users.

7.3.5 Deployment in a Cloud-Secure Container

Although the focus has been on offline systems, there may be scenarios where secure cloud deployment is needed. Containerizing the entire application (e.g., via Docker) with encrypted storage and endpoint control could facilitate deployment in tightly regulated cloud environments.

7.3.6 Domain Adaptation and Fine-Tuning

To improve domain-specific performance (e.g., legal, medical, academic), the system could be fine-tuned on custom corpora using retrieval-augmented fine-tuning strategies. This would enhance both the precision of chunk retrieval and the contextual richness of generated answers.

7.4 FINAL REMARKS

This thesis demonstrated that with thoughtful architectural choices and appropriate use of NLP and vector search technologies, it is possible to design an intelligent, secure, and responsive chatbot system for confidential document access. It addresses key challenges in both AI ethics and enterprise security by allowing organizations to leverage powerful language models without compromising data integrity or user privacy.

The project lays the groundwork for future systems that are not only smarter but also safer, capable of interpreting human language, protecting sensitive data, and adapting to the complex

needs of modern institutions. As language models continue to evolve, combining their power with responsible design practices will define the next era of human-AI interaction.

This research stands as a blueprint for students, researchers, and practitioners striving to integrate natural language technologies into secure real-world applications, efficiently, ethically, and at scale.

References

[1] Oye, E., & Parker, L. (2024). The Role of Natural Language Processing in IoT Chatbot Development.

[2] Luca, C. (2025). Natural Language Processing (NLP) for Document Analysis.

[3] Haider, K. (2024). Natural Language Processing in AI-Powered Systems: Techniques and Future Prospects. *Journal of AI Range*, *1*(1), 40-53.

[4] Borjali, A., Magnéli, M., Shin, D., Malchau, H., Muratoglu, O. K., & Varadarajan, K. M. (2021). Natural language processing with deep learning for medical adverse event detection from free-text medical narratives: A case study of detecting total hip replacement dislocation. *Computers in biology and medicine*, *129*, 104140.

[5] Casey, A., Davidson, E., Poon, M., Dong, H., Duma, D., Grivas, A., ... & Alex, B. (2021). A systematic review of natural language processing applied to radiology reports. *BMC medical informatics and decision making*, *21*(1), 179.

[6] Kovačević, A., Konjović, Z., Milosavljević, B., & Nenadic, G. (2012). Mining methodologies from NLP publications: A case study in automatic terminology recognition. *Computer Speech & Language*, *26*(2), 105-126.

[7] Grouin, C., Grabar, N., Claveau, V., & Hamon, T. (2019, August). Clinical case reports for NLP. In *BioNLP 2019-18th ACL Workshop on Biomedical Natural Language Processing* (pp. 273-282). ACL.

[8] Upadhyaya, N., Joshi, H., & Agrawal, C. (2025). Examining NLP for Smarter, Data-Driven Healthcare Solutions. In *Intelligent Systems and IoT Applications in Clinical Health* (pp. 393-420). IGI Global.

[9] Chaurasia, S. H. U. B. H. A. M., Jain, S. H. U. B. H. A., Vishwkarma, H. O., & Singh, N. I. S. H. A. N. T. (2023). Conversational AI Unleashed: A Comprehensive Review of NLP-Powered Chatbot Platforms. *Iconic Research and Engineering Journals*, *7*(3), 1-8.

[10] Ojha, R. From Algorithms to Conversations: The Influence of Natural Language Processing on Chatbot Innovation.

[11] Babu, A., & Boddu, S. B. (2024). Bert-based medical chatbot: Enhancing healthcare communication through natural language understanding. Exploratory research in clinical and social pharmacy, 13, 100419.

[12] Fu, J., Wang, N., Cui, B., & Bhargava, B. K. (2021). A practical framework for secure document retrieval in encrypted cloud file systems. *IEEE Transactions on Parallel and Distributed Systems*, *33*(5), 1246-1261.

[13] Handa, R., Krishna, C. R., & Aggarwal, N. (2019). Document clustering for efficient and secure information retrieval from cloud. *Concurrency and Computation: Practice and Experience*, *31*(15), e5127.

[14] Li, J. S., Liu, I. H., Tsai, C. J., Su, Z. Y., Li, C. F., & Liu, C. G. (2020). Secure content-based image retrieval in the cloud with key confidentiality. *IEEE Access*, *8*, 114940-114952.

[15] Zhao, Y. S., & Zeng, Q. A. (2018). Secure and efficient product information retrieval in cloud computing. *IEEE Access*, *6*, 14747-14754.

[16] Roshdi, A., & Roohparvar, A. (2015). Information retrieval techniques and applications. *International Journal of Computer Networks and Communications Security*, *3*(9), 373-377.

[17] Ray, A., & Bala, P. K. (2019). Predicting user motivation towards retention of e-services: An NLP-based approach. *International Journal of Business and Administrative Studies*, *5*(1), 1.

[18] Ray, A., Bala, P. K., & Dwivedi, Y. K. (2022). Exploring barriers affecting eLearning usage intentions: an NLP-based multi-method approach. *Behaviour & Information Technology*, *41*(5), 1002-1018.

[19] Tyagi, N., & Bhushan, B. (2023). Demystifying the role of natural language processing (NLP) in smart city applications: background, motivation, recent advances, and future research directions. *Wireless personal communications*, *130*(2), 857-908.

[20] Shahbazi, Z., & Byun, Y. C. (2021). Blockchain-based event detection and trust verification using natural language processing and machine learning. *IEEE Access*, *10*, 5790-5800.

[21] Chowdhury, S., & Nath, A. (2021). Trends in natural language processing: Scope and challenges. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *7*(6), 393-401.

[22] Rocca, R., Tamagnone, N., Fekih, S., Contla, X., & Rekabsaz, N. (2023). Natural language processing for humanitarian action: Opportunities, challenges, and the path toward humanitarian NLP. *Frontiers in big Data*, *6*, 1082787.

[23] Tsarapatsanis, D., & Aletras, N. (2021). On the ethical limits of natural language processing on legal text. *arXiv preprint arXiv:2105.02751*.

[24] Yadav, A., Patel, A., & Shah, M. (2021). A comprehensive review on resolving ambiguities in natural language processing. *AI Open*, *2*, 85-92.

[25] Mahany, A., Khaled, H., Elmitwally, N. S., Aljohani, N., & Ghoniemy, S. (2022). Negation and speculation in NLP: a Survey, Corpora, methods, and applications. *Applied Sciences*, *12*(10), 5209.

[26] Liu, D., Li, Y., & Thomas, M. A. (2017). A roadmap for natural language processing research in information systems.

[27] Arslan, M., Riaz, Z., & Cruz, C. (2023). Revolutionizing management information systems with natural language processing for digital transformation. *Procedia Computer Science*, *225*, 2835-2844.

[28] He, J., Nguyen, D. Q., Akhondi, S. A., Druckenbrodt, C., Thorne, C., Hoessel, R., ... & Verspoor, K. (2021). Chemu 2020: Natural language processing methods are effective for information extraction from chemical patents. *Frontiers in Research Metrics and Analytics*, *6*, 654438.

[29] Majhadi, K., & Machkour, M. (2021). The history and recent advances of natural language interfaces for databases querying. In *E3S web of conferences* (Vol. 229, p. 01039). EDP Sciences.

[30] Makhkamova, O., & Kim, D. (2021). A conversation history-based Q&A cache mechanism for multi-layered chatbot services. *Applied Sciences*, *11*(21), 9981.

[31] Al-Amin, M., Ali, M. S., Salam, A., Khan, A., Ali, A., Ullah, A., ... & Chowdhury, S. K. (2024). History of generative Artificial Intelligence (AI) chatbots: past, present, and future development. *arXiv preprint arXiv:2402.05122*.

[32] Mansurova, A., Nugumanova, A., & Makhambetova, Z. (2023). Development of a question answering chatbot for blockchain domain. *Scientific Journal of Astana IT University*, 27-40.

[33] Aisha, M. A., & Jamei, R. B. (2025). Conversational AI Revolution: A Comparative Review of Machine Learning Algorithms in Chatbot Evolution. *East Journal of Engineering*, *1*(1), 1-18.

[34] Ji, X., Shen, H. W., Ritter, A., Machiraju, R., & Yen, P. Y. (2019). Visual exploration of neural document embedding in information retrieval: Semantics and feature selection. *IEEE transactions on visualization and computer graphics*, *25*(6), 2181-2192.

[35] Webler, J. R. (2019). *a semantic search engine for tagged artworks based on word embeddings* (Doctoral dissertation, Master's Thesis. Ludwig-Maximilians-Universität München).

[36] Monir, S. S., Lau, I., Yang, S., & Zhao, D. (2024). VectorSearch: Enhancing Document Retrieval with Semantic Embeddings and Optimized Search. *arXiv preprint arXiv:2409.17383*.

[37] Yan, F., Fan, Q., & Lu, M. (2018). Improving semantic similarity retrieval with word embeddings. *Concurrency and Computation: Practice and Experience*, *30*(23), e4489.

[38] Rao, R. V., & Selvamani, K. (2015). Data security challenges and its solutions in cloud computing. *Procedia Computer Science*, *48*, 204-209.

[39] Portovaras, T., Kocherov, M., Diegtiar, O., Kizyma, V., & Bakay, V. (2024). Ensuring confidentiality and data security in economic analysis of business entities: Challenges and solutions. *Multidisciplinary Reviews*, *7*(10), 2024245-2024245.

[40] Ibrahim, A. M., Abdel-Aziz, H. R., Mohamed, H. A. H., Zaghamir, D. E. F., Wahba, N. M. I., Hassan, G. A., ... & Aboelola, T. H. (2024). Balancing confidentiality and care coordination: challenges in patient privacy. *BMC nursing*, *23*(1), 564.

[41] Jayasundara, S. H., Gamagedara Arachchilage, N. A., & Russello, G. (2024). SoK: Access Control Policy Generation from High-level Natural Language Requirements. *ACM Computing Surveys*, *57*(4), 1-37.

[42] Abdelgawad, M., Ray, I., Alqurashi, S., Venkatesha, V., & Shirazi, H. (2023, May). Synthesizing and analyzing attribute-based access control model generated from natural

language policy statements. In *Proceedings of the 28th ACM Symposium on Access Control Models and Technologies* (pp. 91-98).

[43] Lee, H. A., Prathapani, N., Paturi, R., Parmaksiz, S., & Di Troia, F. (2022). NLP-based User Authentication through Mouse Dynamics. In *ICISSP* (pp. 696-702).

[44] Weaviate. (n.d.). Mistral – Model providers. Weaviate. https://weaviate.io/developers/weaviate/model-providers/mistral

[45] Chandra, B., Preethika, P., Challagundla, S., & Gogireddy, Y. End-to-End Neural Embedding Pipeline for Large-Scale PDF Document Retrieval Using Distributed FAISS and Sentence Transformer Models. Journal ID, 1004, 1429.

[46] Reddy, N. (2025). Design and Implementation of an AI-Based Chatbot Framework with Retrieval-Augmented Generation and Integrated Recommender System for Interactive User Support. *Available at SSRN 5250507*.

[47] Azhar Sayyad. "A Step-by-Step Guide to Parsing PDFs using the pdfplumber Library." *Medium*, 16 January 2023. https://azhar-sayyad.medium.com/a-step-by-step-guide-to-parsing-pdfs-using-the-pdfplumber-library-in-python-c12d94ae9f07

[48] Bose, P., Roy, S., & Ghosh, P. (2021). A comparative NLP-based study on the current trends and future directions in COVID-19 research. *Ieee Access*, *9*, 78341-78355.

[49] Zaoui Seghroucheni, O., Lazaar, M., & Al Achhab, M. (2025). Using AI and NLP for Tacit Knowledge Conversion in Knowledge Management Systems: A Comparative Analysis. *Technologies*, *13*(2), 87.

[50] Chaid, A. M., Abdulrazzaq, Z. A., Sadoon, R. N., & Aljabery, M. A. (2024). Comparative Analysis of Innovative Machine Learning Algorithms: Advancements in Natural Language Processing.

[51] Upreti, A. (2023). A COMPARATIVE ANALYSIS OF NLP ALGORITHMS FOR IMPLEMENTING AI CONVERSATIONAL ASSISTANTS: Comparative Analysis of NLP Algorithms for NLI.

[52] Zhang, Y., Zhao, X., Wang, Z., & Yu, J. (2022). A hybrid deep learning model for secure document classification and retrieval. *Journal of Information Security and Applications, 65*, 103147.

[53] Ghosal, D., Majumder, N., Poria, S., Gelbukh, A., & Cambria, E. (2020). DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, 154–164.

[54] Bhatia, M., & Gulati, R. (2022). AI-based intelligent document retrieval systems: A review of recent trends and challenges. *International Journal of Computer Applications, 184*(17), 25–32.

Acknowledgement

I would like to express my deepest gratitude to my supervisor, Professor Antonio Roda, for his invaluable guidance, continuous support, and thoughtful mentorship throughout the course of this project. His expertise and encouragement have significantly contributed to the completion and quality of this thesis. I am truly fortunate to have worked under his supervision.

This work is dedicated to my beloved family. My mother, father and sister showed unconditional love and sacrifices. Their constant belief in me has been the foundation of my strength. Their support has been unwavering, and I owe this achievement to their presence in every step of my journey.

I would also like to sincerely thank my friends for their encouragement, understanding, and companionship. Their support has been a steady source of motivation, and their presence has made this journey both manageable and memorable.