InfoGenie: A Chatbot that Enhances Information Extraction Using Modern Natural Language Processing Techniques

Yerram Deekshith Kumar[a], Manash Pratim Lahkar[b], Aditya Kumar Singh[c], Biki Dey[d]
and Utpal Sharma[e]

Department of Computer Science and Engineering, Tezpur University, Napaam, Assam, India

Keywords: Question Answering, Sentence Embeddings, Answer Generation, Information Extraction, PDF Processing.

Abstract: Information extraction and question-answering systems face challenges in efficiently extracting information from large repositories, particularly when dealing with PDF files. In response, this paper presents an innovative application of Natural language processing (NLP) techniques. We address these challenges by developing an intelligent chatbot tailored for streamlined Information extraction. Leveraging established language models and embeddings, including the Hugging Face Transformers library and Sentence Transformer models, our solution seamlessly integrates with the Chroma vector store. We outline a robust data ingestion process encompassing Portable Document Format(PDF) document parsing, text segmentation, and document embedding creation. These embeddings serve as the foundation for a resilient vector store, enhancing Information extraction efficiency. The chatbot's underlying model is fine-tuned for sequence-to-sequence learning, enabling it to generate coherent responses to user queries. Implemented through a user-friendly web interface powered by Streamlit, users can interact seamlessly with the chatbot, upload PDF documents, and ask queries based on those PDF documents. Evaluation on a crowdsourced dataset collected by us demonstrates a 95% cosine similarity between generated and ground truth answers. This research advances NLP-based Information extraction systems, offering practical solutions and insights for future enhancements.

1 INTRODUCTION

The volume and diversity of textual material in the digital world has risen considerably in recent years, and an increasing number of papers in various forms are being generated and exchanged. This flow of data presents possibilities and problems, particularly in the fields of information extraction and question-and-answer systems. Despite the abundance of data that might give insightful knowledge, users may find it challenging to locate and extract the specific information they seek owing to the volume and variety of documents.

Traditional keyword-based search engines and information extraction systems, although occasionally successful, usually struggle to understand the nuances of human language and accurately interpret user requests. The proliferation of unstructured material, such as PDF documents, exacerbates the problem, since standard search algorithms struggle to sort through its complexities.

In this research project, we developed an intelligent chatbot capable of understanding user requests and responding in natural language. The project's purpose is to bridge the knowledge gap between consumers and the massive amounts of information concealed behind digital archives. The chatbot's goal is to improve users' access to relevant information by utilizing sophisticated language understanding skills and innovative document processing techniques. This allows consumers to make more educated judgments and gain deeper insights from textual data.

2 APPLICATIONS

Our study has broad applicability in numerous sectors. Some of the main applications are:

[a] https://orcid.org/0009-0001-1647-8522
[b] https://orcid.org/0009-0008-7244-1653
[c] https://orcid.org/0009-0009-3588-1366
[d] https://orcid.org/0009-0000-4342-854X
[e] https://orcid.org/0000-0002-9210-7168

2.1 Information Extraction

- The key use case for our chatbot is information extraction, which allows users to quickly search through a library of papers for relevant information.

- This tool is beneficial in academic contexts, corporate settings, research, and any other circumstance in which quick access to certain information is required.

2.2 Document Summarizing

- Our chatbot can simplify and summarize long-form papers, allowing users to rapidly comprehend the essential arguments and conclusions without having to read the entire thing.

- This tool is beneficial for professionals that need to quickly extract critical data from long reports or research papers.

2.3 Customer Support and Faqs

- Businesses may provide 24-hour customer service by incorporating our chatbot into their customer care systems.

- The chatbot may react to customer queries regarding goods, services, rules, and troubleshooting in a timely and accurate manner, relieving the burden on human support representatives.

2.4 Educational Tools

- In the field of education, our chatbot might function as a virtual study assistant or tutor, assisting students in identifying relevant literature, responding to questions, and offering clarification on a number of issues.

- Furthermore, it may assist teachers in creating interactive lectures and determining how effectively their pupils understand the subject taught in class.

2.5 Legal and Compliance

- The chatbot can be used by law firms and legal departments to look up pertinent precedents, rules, or legal interpretations by searching through statutes, legal documents, and case law.

- Law companies and legal departments can utilize the chatbot to search for relevant precedents, regulations, or legal interpretations in legislation, legal documents, and case law.

Although this kind of automated chatbot seems promise, there are a few issues that need to be resolved. One of the main challenges is creating precise algorithms to examine the intricate structure of digital documents, especially PDF files. Maintaining real-time responsiveness while effectively and scalable managing massive amounts of data is a major technical issue.

3 LITERATURE REVIEW

The following section covers research articles from areas like NLP, Information extraction and Information retrieval:

3.1 Research on Document Similarity, Summarization, and Retrieval

Our chatbot system's primary tasks are retrieval, summarization, and document similarity. Here, we examine a number of studies that investigate methods for these assignments:

Ascione and Sterzi conduct a comparative analysis of embedding models for measuring patent similarity (Ascione and Sterzi, 2023). The selection of an appropriate model for our chatbot to evaluate document relevance can be informed by embedding models, which capture semantic relationships between documents. (Ascione and Sterzi, 2023).

Sequence-to-sequence Recurrent neural networks (RNNs) are the means by which Nallapati et al. propose an abstractive text summarization approach (Nallapati et al., 2016). Abstractive summarization is the process of creating fresh, succinct summaries that highlight the key ideas in a work. Although our first focus is on retrieval, adding abstractive summarization to our chatbot might be a useful addition in the future(Nallapati et al., 2016).

For retrieval tasks, Askari et al. provide an effective transformer-based re-ranker (Askari et al., 2023). Re-ranking is the process of picking the best papers from a set of candidate documents that a previous system had obtained. According to their research, transformer-based models may significantly increase the information extraction accuracy of our chatbot (Askari et al., 2023).

Jiang et al. offer a technique for rating large texts using query-directed sparse transformers (Jiang et al., 2023). Ranking refers to the process of sorting returned documents based on their relevance to

the user's query. Jiang's work on sparse transformers provides excellent ways for handling long documents when our chatbot works with huge datasets (Jiang et al., 2023).

Together, these studies provide innovative approaches for increasing the efficiency and quality of information extraction, which is critical for our chatbot's ability to identify relevant material for users.

3.2 Integration of Language Models and Artificial Intelligence(Ai) in Educational and Enterprise Settings

Several studies explore integrating large language models and AI services into educational and enterprise applications, demonstrating the potential of these technologies in real-world settings:

Hsain and El Housni (Hsain and El Housni, 2023) investigate the use of large language model-powered chatbots to support students in higher education. Their work suggests that large language models can be beneficial for educational chatbots, providing a foundation for our chatbot's ability to interact with users and answer their questions .

Jeong (Jeong, 2023) explores the implementation of generative AI services in enterprise applications. Generative AI models can be used for various tasks, including text generation and chatbot development. Jeong's work highlights the potential for generative AI to enhance the capabilities of enterprise chatbots, providing insights for us to consider as we develop our own chatbot .

Taipalus (Taipalus, 2023) discusses fundamental concepts and challenges associated with vector database management systems, which are essential for storing and retrieving high-dimensional data like document embeddings. Efficient storage and retrieval of document embeddings are crucial for our chatbot's performance. Taipalus's work highlights the importance of considering appropriate data storage solutions for our chatbot .

Shen et al. (Shen et al., 2023) propose a framework for memory augmentation using language models, offering insights for enhancing the chatbot's knowledge retention and retrieval capabilities. Memory augmentation techniques can improve a chatbot's ability to access and process information, potentially benefiting our chatbot's ability to answer follow-up questions and engage in multi-turn conversations .

Van de Cruys et al. (Van de Cruys et al., 2022) investigate question-answering techniques for technical documents. While our initial focus might be on Information extraction, incorporating question-answering capabilities could be a valuable future extension for our chatbot. Van de Cruys et al.'s work provides insights into techniques for enabling our chatbot to answer user questions directly within retrieved documents.

Adiba et al. (Adiba et al., 2023) propose methods for unsupervised domain adaptation in question-answering systems. Unsupervised domain adaptation allows a model to be trained on data from one domain (e.g., general knowledge) and then applied to a different domain (e.g., legal documents) where labeled data is scarce. While our initial focus might be on retrieval, incorporating question-answering capabilities could be a valuable future extension for our chatbot, especially when dealing with domain-specific documents. Adiba et al.'s work suggests that unsupervised domain adaptation techniques could help enable our chatbot to answer questions about these specialized documents even if limited training data is available in that specific domain (Adiba et al., 2023).

Cohan et al. (Cohan et al., 2023) explore the use of pre-trained language models for sequential sentence classification tasks. Sentence classification involves categorizing sentences based on their meaning. While our initial focus might be on retrieval, incorporating functionalities like sentiment analysis or topic classification could be valuable extensions for our chatbot. Cohan et al.'s work suggests that pre-trained language models can be effective for these tasks, providing a foundation for us to explore adding such functionalities in the future .

Kamma (Kamma, 2023) discusses language modeling for intelligent Information extraction systems. Kamma's work emphasizes the role of language models in understanding the semantics of documents and queries, which is essential for effective retrieval. Their insights can inform our selection and application of language models within our chatbot's retrieval system .

Lappalainen and Narayanan (Lappalainen and Narayanan, 2023) describe Aisha, a custom AI library chatbot built using the ChatGPT API. While this work directly utilizes an existing API, it showcases the potential for building custom chatbots with capabilities similar to our envisioned intelligent Information extraction chatbot .

Trust et al. (Trust et al., 2024) explore techniques for augmenting large language models to enhance interaction with government data repositories. While their focus is on a specific domain (government data), their work highlights the potential for ongoing research and development in large language models,

which can inform future advancements in our own chatbot's capabilities .

Vaswani et al. (Vaswani et al., 2017) introduce the Transformer architecture, a neural network architecture that has become foundational for many state-of-the-art NLP models, including some of the works discussed previously (e.g., Askari et al. 2023). Understanding the core principles of the Transformer architecture can provide valuable background knowledge for us as we develop our chatbot.

Wang et al. (Wang et al., 2019) discuss language models with transformers, providing a more in-depth exploration of this architecture and its applications in various NLP tasks. Similar to Vaswani et al. (2017), this work can provide a deeper understanding of the technical foundations underlying some of the recent advancements in NLP relevant to our chatbot development.

In conclusion, this review has explored a wide range of advancements in NLP, Information extraction, and chatbot systems. The referenced studies offer valuable insights into methodologies that can be harnessed to enrich the capabilities of our intelligent Information extraction chatbot. Key takeaways include:

Building an effective chatbot requires mastery of essential information extraction methodologies. Embedding models and abstractive summarization aid in understanding PDF information and condensing large documents into concise, understandable summaries. Transformer-based models improve retrieval, providing for rapid access to important information. Integrating massive language models with AI characteristics such as instructional chatbots and generative AI enhances capabilities for question response and sophisticated information processing. Keeping up with NLP breakthroughs enables constant progress. Understanding Transformer architecture helps you make educated decisions about how to deploy these technologies, such as creating a robust, user-friendly chatbot that successfully answers questions and offers important information.

4 SYSTEM ARCHITECTURE

Our system's functionalities include processing user queries quickly, retrieving pertinent data from a knowledge base, and producing context-aware responses. Figure 1 shows the information flow from user queries to the creation of ranked results, representing the system architecture. Important parts of the architecture include the generative model, knowledge base, text chunking, and embeddings generation. Algorithm 1 describes the system's workflow.
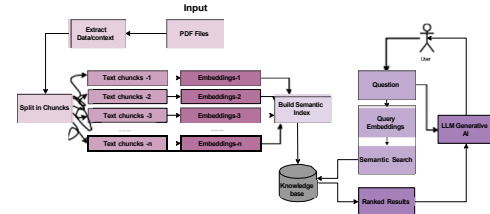
Figure 1: System Architecture.

Algorithm 1 NLP-based Query Chatbot

Require: User query q, PDF documents $D = \{d_1, d_2, ..., d_n\}$
Ensure: Context-aware response r
0: **function** CHATBOT(q, D)
0:     Extract text chunks: $T \leftarrow \{\}$
0:     **for** $d_i \in D$ **do** $T \leftarrow T \cup$ EXTRACTCHUNKS(d_i)
0:     **end for**
0:     Embed chunks: $E \leftarrow \{\}$
0:     **for** $t_j \in T$ **do** $E \leftarrow E \cup$ EMBEDDING(t_j)
0:     **end for**
0:     Create semantic index I from E
0:     Transform query: $q_{embed} \leftarrow$ EMBEDDING(q)
0:     Retrieve relevant embeddings:
       $rel\_emb \leftarrow$ SEMANTICSEARCH(I, q_{embed})
0:     Rank results: $ranked\_res \leftarrow$ RANK(rel_emb)
0:     Generate response:
       $r \leftarrow$ GENERATERESPONSE($ranked\_res$)
0:     **return** r
0: **end function**=0

4.1 SYSTEM COMPONENT

The system is made up of a number of interrelated components, each of which plays an important part in obtaining chatbot functionality. Table 1 presents an overview of these components and their interconnections.

4.2 Components of the Language Model Pipeline

The Language Model Pipeline consists of following key components:

• **MiniLM Model**: HuggingFace Transformers' MiniLM model, a transformer-based language model, serves as the pipeline's primary component. MiniLM was chosen because it excels in understanding the context of user requests and providing well-reasoned responses. The model understands complicated linguistic patterns since it has been pre-trained on a large corpus.

Table 1: System Components Overview.

Component	Description
User	Initiates queries and interacts with the chatbot.
Question	User's input in the form of a query.
Text Chunks	Segments of text extracted from PDF files.
Embeddings	Numerical representations of text chunks.
PDF Files	Source documents containing relevant information.
Knowledge Base	Repository of text chunks and embeddings.
LLM Generative Model	Language model for context-aware response generation.
Semantic Search	Process of retrieving information based on semantic similarity.
Ranked Results	Ordered list of relevant results.

- **Tokenizer**: Tokenization is the process of transforming input text into units that the MiniLM model understands. This pipeline uses HuggingFace's `AutoTokenizer`, making it straightforward to load the MiniLM model's pre-trained tokenizer.

- **Pipeline Parameters**: A variety of settings may be configured to optimize the behavior and performance of the Language Model Pipeline. Table 2 summarizes the parameters utilized in the Language Model Pipeline.

Table 2: Language Model Pipeline Parameters

Parameter	Description
Model	MiniLM
Max Length	256
Sampling	True
Temperature	0.3

5 METHODOLOGY

The procedures for text preparation, indexing, query processing, and result display are as follows:

5.1 TEXT PREPROCESSING AND REPRESENTATION

PDF files are processed using PyPDFLoader and PDFMinerLoader. Textual data is extracted from PDF files. The retrieved text has been divided into manageable chunks.

Let D be the collection of PDF documents, and let T_i be the set of text excerpts extracted from each document d_i in D. To extract the associated text chunks T_i, each document d_i in D is iterated over and analysed individually. It is possible to put this this way:

$$T_i = \text{ExtractChunks}(d_i) \tag{1}$$

Sentences are added to the text chunks T_i once they have been collected for each document d_i. Let S_i be the collection of sentences that were taken out of each of the text chunks T_i. To do this, each text chunk t_j in T_i is divided into separate sentences s_k. This might be shown as:

$$S_i = \bigcup_{t_j \in T_i} \{s_k \mid s_k \text{ is a sentence in } t_j\} \tag{2}$$

After obtaining the collection of sentences S_i for each document d_i, embeddings are created for each sentence to aid in semantic analysis. E_i should be the collection of embeddings corresponding to the sentences in S_i. As part of the embedding process, each phrase s_k in S_i is iterated over, and the appropriate embedding e_k is calculated. This can be displayed as:

$$E_i = \{e_k \mid s_k \in S_i\} \tag{3}$$

To generate a semantic index I for efficient information extraction, each document d_i has its embeddings E_i used. Finally, the text segments, phrases, and embeddings in each document are ready to be employed in responses to user queries.

5.2 INDEXING AND STORAGE

We save the sentence embeddings with a semantic index (e.g., Chroma). The accuracy and speed of information extraction are improved by integrating with Chroma. In order for the system to efficiently and correctly retrieve data based on the semantic knowledge embedded in the vectors, the vector storage is essential to similarity searches.

5.3 QUERY PROCESSING

To get ready for processing, a user query q is tokenized when it is received. Let $q_{embedding}$ stand for the query's embedding, which was acquired by using the same procedure as the text chunks. To obtain relevant results, *relevant_embeddings*, a semantic search is run on the stored embeddings in Chroma.

5.4 RESULT PRESENTATION

Based on how well the returned results match the query semantically, they are ordered. The ordered list of pertinent results from the semantic search

may be represented as follows: *ranked_results* = {r_1, r_2, ..., r_k}, where r_i indicates the *i*-th result. A tuple (*text_i*, *score_i*), with *text_i* denoting the result's text content and *score_i* denoting its semantic similarity score with the query, makes up each result r_i.

The last response is produced using a Language Model (LM) Generative AI. The function that produces the response based on the query *q* and the ranked results is denoted by *A*(*q*, *ranked_results*). The following yields the answer *a*:

$$a = \text{Restructure}\left(\text{argmax}_{r_i \in \text{ranked\_results}} \text{Score}(r_i)\right) \quad (4)$$

In this notation:

• *a* represents the final answer.

• The function argmax selects the result r_i with the highest score from the ranked results.

• *Score*(r_i) denotes the score assigned to each result r_i.

• *Restructure* represents the function that restructures the selected answer before presenting it as the final response.

5.5 DATASET

We decided to use Google Forms for our assessment dataset annotations, employing a crowdsourcing approach. Participants received the PDF documents for the evaluation task and were asked to come up with questions and answers based on the content. The goal was to directly collect these annotations through the form.In order to guarantee that the annotations were excellent and pertinent, we took care to specify standards including clarity, correctness, and alignment with the PDF text.
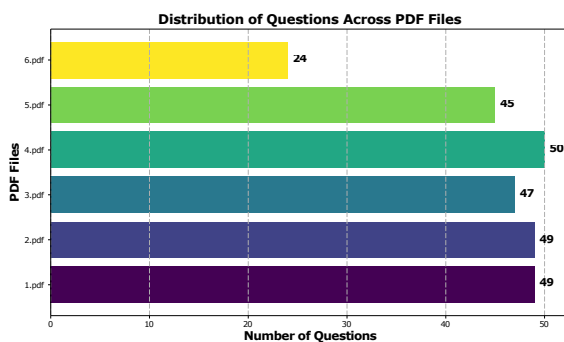

Figure 2: Count of questions for each document

Six PDF files make up our dataset, which has 2,295 tokens and 982 unique words in total. Each paragraph has around 382.5 tokens on average, dispersed over 14 phrases, or roughly 27.32 tokens per sentence. We collected 264 submissions in all, with 44 questions on average per PDF file and responses that were around 18.14 words long. We carefully examined and verified the annotations during the data gathering procedure to ensure their validity. The evaluation dataset[1] was enhanced with a multitude of viewpoints and insights by our crowdsourcing technique, which was made possible by the wide range of questions and answers from human annotators. We intend to increase the scope of our crowdsourced dataset by include a greater variety of document formats and query types, in recognition of its limitations. This will entail gathering information from a variety of sources and making sure that our dataset accurately reflects a range of real-world circumstances. Our goal is to increase the assessment dataset's diversity in order to enhance the chatbot's resilience and generalizability when processing various kinds of documents and questions.

6 EXPERIMENTS AND RESULTS

We conduct a comprehensive investigation of the chatbot's performance in the Colab environment using scores from the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) and a range of word embedding models to determine how effectively the bot understands user queries and generates relevant responses.

Table 3 shows that the average cosine similarity scores for the various embeddings (GloVe, Word2Vec, FastText, and BERT) vary from 0.82 to 0.95. The produced responses' semantic closeness to the ground truth answers may be inferred from these ratings. We use four different embeddings: GloVe (glove-wiki-gigaword-300), Word2Vec (word2vec-google-news-300), FastText (fasttext-wiki-subwords-300), and BERT (bert-base-uncased) to turn the ground truth response and the anticipated answer into vectors.

Following conversion, the similarity between the ground truth answer's and the anticipated answer's vectors was measured using cosine similarity. Fig. 3 displays the document-wise cosine similarity of the anticipated responses with the ground truth.

Additionally, we evaluated the chatbot's quality of answer using ROUGE scores, a widely used natural language processing metric. The following were the acquired ROUGE scores:

• **ROUGE-1:** Recall (r) = 0.54, Precision (p) = 0.44, F1-score (f) = 0.46

---

[1]https://drive.google.com/drive/folders/1B$_H$$Uw0u2$ *jb* 9*f CIS5RWJcrweD9s4KSAgt?usp = sharing*

Table 3: Average Cosine Similarity Scores for Different Embeddings

Doc No.	Embeddings			
	glove-wiki -gigaword-300	word2vec -google-news-300	fasttext-wiki -subwords-300	bert-base -uncased
1.pdf	0.84	0.73	0.91	0.84
2.pdf	0.92	0.78	0.94	0.90
3.pdf	0.95	0.89	0.97	0.97
4.pdf	0.94	0.84	0.96	0.94
5.pdf	0.92	0.84	0.96	0.95
6.pdf	0.97	0.92	0.98	0.97
Average	0.92	0.82	0.95	0.92

- **ROUGE-2:** Recall (r) = 0.31, Precision (p) = 0.28, F1-score (f) = 0.28

- **ROUGE-L:** Recall (r) = 0.48, Precision (p) = 0.40, F1-score (f) = 0.42

The reported ROUGE scores show that the chatbot's capacity to create replies with significant lexical and semantic overlap with ground truth answers still needs to be improved. Additional improvements are required to improve the relevancy and accuracy of the chatbot's answers. By combining the ROUGE score evaluation with the examination of average cosine similarity scores, we can gain a comprehensive understanding of the chatbot's ability to answer to user inquiries in a relevant and accurate manner. These measures are useful for determining the quality and usefulness of a chatbot's replies in real-world circumstances.
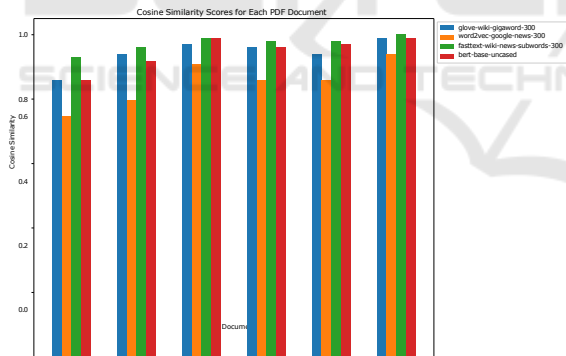

Figure 3: Document wise average Cosine similarity of the answers.

7 WEB-BASED USER INTERFACE FOR ENHANCED INTERACTION

This section introduces our chatbot application's web-based user interface, which is intended for interactive information retrieval and question-answering. Streamlit and simple HTML templates were used in the interface's development to ensure that users could easily query information from documents.
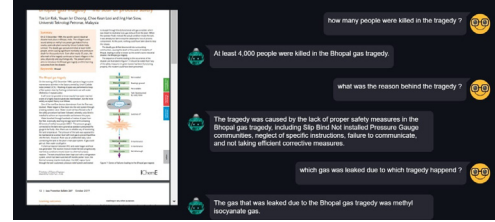

Figure 4: User Interface of Chatbot Web Application.

7.1 Article Display

The article about the Bhopal gas tragedy is displayed on the left side of the interface. The causes of the incident, the number of casualties, and other relevant details are all covered in detail in this article. Users can consult the original content while interacting with the chatbot because the article is presented in an easily readable format.

7.2 Chatbot Interaction Panel

Users can interact with the chatbot through the chat interface on the right side. Users are intended to ask questions about the article's content using this panel. Utilising sophisticated natural language processing methods, the chatbot retrieves pertinent data from the document in response to user inquiries.

7.3 Illustrative Interaction

A typical user interface interaction is shown in Figure 4. In this instance, the user poses particular queries regarding the Bhopal gas tragedy, and the chatbot provides precise answers taken directly from the article. The exchange shows off the chatbot's abilities to:

- Accurately identify and retrieve key details such as the number of casualties.

- Explain the reasons behind the tragedy, including safety failures and procedural neglect.

- Specify the type of gas involved in the incident.

The seamless user experience is made possible by the interface's aesthetic appeal and functionality, which are ensured by the integration of HTML templates and Streamlit. We carried out official user testing sessions to make sure the interface satisfies user expectations and demands. A wide range of individuals engaged with the chatbot throughout these ses-

sions, offering qualitative input on the usability, navigation, and general performance of the interface. The input received was crucial in directing more improvements. Our system serves as an example of how contemporary chatbots can greatly improve information extraction and retrieval, enabling users to more effectively extract specific details from lengthy texts.

7.4 Proof of Concept

It is important to note that this interface is intended as a proof of concept. While it demonstrates the potential capabilities and effectiveness of the chatbot, further development and refinement would be necessary for practical deployment and broader application across different domains and types of documents.

8 LEGAL AND ETHICAL ISSUES

The use of automated information extraction and question-answering systems in sensitive sectors presents significant legal and ethical concerns. These systems must adhere to privacy standards, ensuring that consumers are aware of data usage and provide explicit consent. Data anonymization and encryption are critical for ensuring user privacy. Furthermore, NLP models might perpetuate biases found in their training data, resulting in biassed outputs. To limit this risk, employ diversified datasets, fairness-aware algorithms, and conduct frequent bias checks. Transparency helps foster trust by helping users to understand how decisions are made, and developers must accept responsibility for the system's consequences, including methods for fixing failures. Automated technologies should support rather than replace human decision-making, especially in vital sectors such as healthcare and finance. Ethical norms must be set to avoid misuse and promote beneficial societal consequences. Addressing these legal and ethical concerns is critical for the proper development and deployment of NLP-based information extraction systems, which will increase their value and adoption in real-world applications.

9 CONCLUSION AND FUTURE WORKS

In conclusion, our research demonstrates a signifi- cant advancement in natural language processing and Information extraction through the development of an

effective chatbot. Employing various word embedding models and evaluating the chatbot's performance using average cosine similarity scores and ROUGE scores, we have shown its ability to comprehend user queries and provide accurate responses from a document corpus. Despite constraints on directly applying traditional metrics like precision and recall, our focus remains on optimizing user experience and facilitating meaningful interactions. The analysis of average cosine similarity scores across different embeddings provides insights into the semantic similarity between generated responses and ground truth answers. Furthermore, ROUGE scores offer valuable indications of lexical and semantic overlap, contributing to our understanding of the chatbot's performance. These findings underscore the chatbot's efficacy in generating relevant and accurate responses, laying a strong foundation for future advancements in intelligent document querying systems.

Looking ahead, future enhancements aim to enhance the system's power and adaptability. Integrating advanced language models, exploring diverse vector store options, and extending features to support multimedia content will contribute to the continuous evolution of the chatbot.

A significant avenue for future development involves customizing the system to be organization-specific. Tailoring the chatbot to address the unique queries and requirements of different organizations aims to provide a valuable tool for enhancing Information extraction in professional settings.

In summary, the chatbot not only achieves its primary goal of efficient Information extraction but also lays the groundwork for further advancements in natural language processing. This project serves as a foundation for ongoing research and application development, contributing to the continuous evolution of intelligent systems for document querying.

9.1 Future Directions

We will execute domain-specific assessments in the financial, legal, and healthcare sectors to give a thorough study of the chatbot's performance across several domains. This will evaluate how well the chatbot can handle language unique to its domain, adapt, and deliver trustworthy information. We will also assess the system's performance in managing massive amounts of data and real-time processing, addressing scalability and efficiency issues. We will investigate optimisation strategies including model compression and parallel processing. Formal user testing sessions will be carried out to improve the user experience by collecting qualitative input on the usability and in-

terface of the chatbot, which will direct future improvements. Finally, by applying sophisticated natural language understanding algorithms and context management strategies, we will handle technical issues including managing complicated queries, document parsing failures, and keeping context throughout multi-turn interactions. With these improvements, InfoGenie should be a more reliable and adaptable tool for automated information extraction, increasing its wider application and practicality.

REFERENCES

Adiba, A. I., Homma, T., and Sogawa, Y. (2023). Unsupervised domain adaptation on question-answering system with conversation data.

Ascione, G. S. and Sterzi, V. (2023). A comparative analysis of embedding models for patent similarity.

Askari, A., Verberne, S., A bolghasemi, A., Kraaij, W., and Pasi, G. (2023). Retrieval for extremely long queries and documents with rprs: a highly efficient and effective transformer-based re-ranker.

Cohan, A., Beltagy, I., King, D., Dalvi, B., and Weld, D. S. (2023). Pretrained language models for sequential sentence classification.

Hsain, A. and El Housni, H. (2023). Large language model-powered chatbots for internationalizing student support in higher education.

Jeong, C. (2023). A study on the implementation of generative ai services using an enterprise data-based llm application architecture.

Jiang, J.-Y., Xiong, C., Lee, C.-J., and Wang, W. (2023). Long document ranking with query-directed sparse transformer.

Kamma, A. (2023). An approach to language modelling for intelligent document retrieval system.

Lappalainen, Y. and Narayanan, N. (2023). Aisha: A custom ai library chatbot using the chatgpt api.

Nallapati, R., Zhou, B., dos Santos, C., Gu¨lc¸ehre, , and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. IBM Watson.

Shen, J., Dudley, J. J., and Kristensson, P. O. (2023). Encode-store-retrieve: Enhancing memory augmentation through language-encoded egocentric perception.

Taipalus, T. (2023). Vector database management systems: Fundamental concepts, use-cases, and current challenges.

Trust, P., Omala, K., and Minghim, R. (2024). Augmenting large language models for enhanced interaction with government data repositories. University College Cork, Cork, Ireland.

Van de Cruys, T., Vanroy, B., and Peirsman, Y. (2022). Question answering on technical documents.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, , and Polosukhin, I. (2017). Attention is all you need. Google Research.

Wang, C., Li, M., and Smola, A. J. (2019). Language models with transformers. Amazon Web Services.