Empowering Enterprise Architecture: Leveraging NLP for Time Efficiency and Strategic Alignment

Sunidhi Goyal, Fadumo Hussein, Isha Thukral, Taryn Trimble
School of Data Science
University of Virginia

***Abstract*—In 2014, the U.S. legislation enacted the Federal Information Technology Acquisition Reform Act which establishes requirements and guidelines that indirectly impact enterprise architecture within federal agencies. These regulations emphasize the importance of strategic IT planning, effective governance, and alignment of IT investments with organizational goals, which are key aspects of enterprise architecture. Enterprise architects play a critical role in shaping organizational structures and processes, but this is coupled with the increasing demands on their time. This project addresses the need to alleviate the time constraints faced by enterprise architects by introducing an NLP-based Enterprise Architecture Assistant (EAA). By harnessing NLP capabilities, the EAA aims to streamline various tasks associated with enterprise architecture, such as documentation, analysis, and decision-making processes, which are traditionally managed manually by enterprise architects. It utilizes NLP for filtering documents of relevance, and their subsequent summarization and employs radial graph visualization to develop an intuitive sense for the enterprise architect to observe mentioned connections in the document. These methods expedite enterprise architects' tasks by extracting key insights from documents and presenting them visually. The EAA aims to streamline decision-making and documentation processes, reducing the time burden on architects. Overall, the project's methodology harnesses NLP's capabilities to create a user-friendly toolset for managing enterprise architecture efficiently. Through the application of these techniques, the project aims to achieve several key results. The automated process of filtering and retaining relevant documents from a larger set is enhanced and leaves room for the enterprise architect to focus on important tasks, and boosts organizational productivity. Furthermore, implementing a customised chatbot trained on the filtered documents enhances functionality to facilitate user interaction and information retrieval from the architectural docs. The chatbot serves as an intuitive interface for accessing and querying the data. Visual representations provide insights into the connections and dependencies within the documents, thereby enhancing comprehension. These results hold the potential to revolutionize enterprise architecture practices, fostering efficiency, agility, and collaboration across organizational initiatives in alignment with organizational priorities.**

***Index Terms*—Enterprise architecture, NLP, Decision-making, ChatBot, Efficiency, Optimization.**

I. INTRODUCTION

Efficiently managing and extracting value from large digital document repositories has become a significant challenge in the fast-developing subject of enterprise architecture (EA). In order to increase enterprise architects' productivity, this project presents the Enterprise Architecture Assistant (EAA), a cutting-edge tool that makes use of sophisticated natural language processing (NLP) techniques [1]. The EAA tackles two primary issues: the need for an effective, interactive way to interact with EA data and models, and the time-consuming task of sifting through copious documentation to locate pertinent information. Through the integration of a chatbot for interactive questions and the automation of document relevancy filtering, the EAA seeks to improve accessibility to architectural knowledge for all businesses while also streamlining architectural design processes.

This paper discusses the development and components of the EAA—comprising a document relevance checker, an NLP-enabled chatbot, and a visualization interface for EA models—emphasizing its potential to revolutionize EA practices by improving efficiency and strategic focus. The evaluation of the EAA demonstrates its capability to significantly aid enterprise architects in navigating and analyzing complex documentation, fostering organizational development and transformation.

II. BACKGROUND AND MOTIVATION

The primary purpose of the project is to reduce a considerable amount of time spent by the Enterprise Architecture team in an organization while designing frameworks of team functionalities, within the organization. This would provide enterprise architects with more time to work on strategic tasks. We aim to optimize the time spent using NLP techniques, by helping them sift through a myriad of documents through our model and only retain the relevant documents that are necessary for the process. Furthermore, to summarize concepts that lay in a vast sea of those documents quickly, we aim to build a retrieval augmented generation ChatBot and provide the team with important visualizations that depict the level of contingency of different teams on each other.

III. METHODOLOGY

The envisioned pipeline is architected to encompass three distinct modules: A relevance checker, chatbot, and a user interface that visualizes the relationship between the mentioned stakeholders in the document. Users are afforded the capability to submit a collection of documents in PDF format, which the pipeline will process. The relevance checker evaluates the pertinence of the documents by analyzing the similarity of topics discussed therein. Concurrently, the chatbot is equipped to provide summaries and respond to inquiries regarding the documents' content. This is facilitated through an intuitive user interface, which also enables users to examine the interconnections between the stakeholders mentioned within a specific document in a visual format. This comprehensive system aims to streamline the analysis of extensive document collections, ensuring efficient information retrieval and stakeholder relationship mapping. This pipeline not only simplifies the user's navigation through the document corpus but also enhances the comprehensibility of stakeholder interconnections. Collectively, these components form a cohesive system that aims to streamline the processing and analysis of large volumes of PDF-formatted documents, ensuring that users can efficiently extract and visualize pertinent information and stakeholder dynamics.

A. *Data Acquisition*

In this study, we employed a two-pronged approach for data acquisition, encompassing both web scraping and simulation techniques to gather relevant enterprise architecture books and documents, as well as to generate synthetic data for various stages of the pipeline. To collect real-world enterprise architecture materials, we conducted targeted web scraping from reputable online repositories, academic databases, and digital libraries [2]. Leveraging custom-built web scrapers, we systematically retrieved documents, whitepapers, and scholarly articles that cover topics related to enterprise architecture, strategic IT planning, governance, and organizational alignment. This process ensured a diverse and comprehensive dataset representing contemporary practices and perspectives in the field. Additionally, we acquired a collection of enterprise architecture documents in PDF format from industry reports and academic publications. Utilizing PDF parsing tools, we converted these PDF files into machine-readable text format. This step facilitated the integration of PDF-based content into our dataset, ensuring compatibility with the NLP-based analysis and processing pipeline. Furthermore, complementing the acquired real-world data, we simulated synthetic data to augment our dataset and enrich the training pipeline. Through carefully crafted algorithms and models, we generated data instances representing various scenarios encountered in enterprise architecture tasks, including but not limited to decision-making processes, documentation formats, and stakeholder interactions. This synthetic data injection enabled us to validate the robustness and adaptability of our NLP-based Enterprise Architecture Assistant (EAA) across a spectrum of simulated conditions, thereby enhancing the generalizability of our findings.

B. *Preprocessing*

The design of this pipeline is predicated on the principle of maximizing user efficiency, with a particular emphasis on minimizing the temporal expenditure associated with each document's analysis. The architecture of the system necessitates little preprocessing, which makes document integration quick and easy. The main requirement for documents going through the pipeline is that they include a lot of text content that is easily readable. This criterion is important since text translation from non-textual materials like photographs or graphical representations is outside the scope of the pipeline's existing technological capabilities. Consequently, documents with large textual content are more suited to the pipeline's processing algorithms, enabling a full and comprehensive study.

C. *Topic Modeling using PCA & LDA*

To effectively identify relevant documents in our study of enterprise architecture, we employed two main methods: Principal Component Analysis (PCA) [3] and Latent Dirichlet Allocation (LDA) [4]. The purpose of using these specific clustering techniques was to organise the document corpus according to the interests of the enterprise architect. This strategic action is intended to eliminate the need for the architect to perform a thorough analysis of an enormous number of documents. This ensures that the architect can quickly locate and retrieve papers related to the particular topic under examination. This methodological approach optimises the architect's research efficacy by greatly streamlining the information retrieval procedure and improving the relevance of the search results.

1) Principal Component Analysis: To isolate and classify pertinent documents, we used Principal Component Analysis (PCA) in our study to cluster enterprise architecture documents according to their topical coverage. PCA is a popular data analysis technique that helps to reduce the dimensionality of complicated datasets while maintaining important information. PCA finds the principle components that encapsulate the maximum variation in the dataset by converting the initial high-dimensional data into a lower-dimensional representation.

Principal Component Analysis (PCA) was carried out in our research using Gensim, a Python module that is frequently used for tasks related to natural language processing, such as document clustering and topic modelling [5]. A PCA solution designed especially for text data, Gensim enables the investigation of latent structures and patterns in textual corpora. First, the textual data is represented in a numerical format that can be analysed using Gensim's PCA technique. In order to do this, the texts are usually converted into vector representations, like bag-of-words or TF-IDF vectors [6], in which each dimension is mapped to a distinct vocabulary phrase. Gensim's Principal Component Analysis (PCA) efficiently decreases the dimensionality of the dataset while maintaining its key features by keeping just the top principal components that contribute significantly to the variance [7]. As a result, each document is now represented by a smaller collection of principle components, resulting in a lower-dimensional representation of the textual data.

PCA streamlined clustering by simplifying feature extraction and eliminating irrelevant data. It facilitated separating clusters by topic and identifying similar documents, improving overall clustering accuracy. This enabled the isolation of pertinent documents from the dataset, enhancing subsequent chatbot training. Moreover, PCA's ability to discern underlying data variation allowed for clustering by multiple topical groups. As depicted in Figure 1, PCA effectively grouped enterprise architecture documents, allowing for prioritization of relevant texts in future analyses.

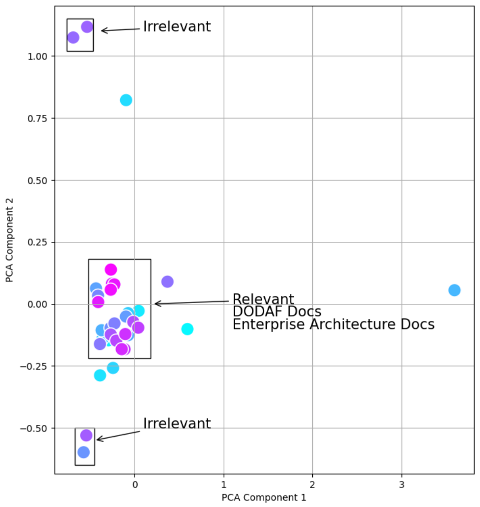


Fig. 1: PCA Document Analysis

To note our study's focus has been narrowed to focus on the enterprise architecture document cluster as a result of the PCA document analysis. Because of this, we have eliminated papers that PCA deemed unnecessary, guaranteeing a focused and effective study that is especially relevant to the field of enterprise architecture.

2) Latent Dirichlet Allocation: Using the Gensim library, we performed Latent Dirichlet Allocation (LDA) similarity and dissimilarity analysis in our study. The measurement of document similarities and dissimilarities as well as the discovery of latent topics inside textual corpora were made possible by LDA, a probabilistic generative model that is frequently used in natural language processing [4]. Within the LDA framework, similarity analysis required determining how similar two documents were to one another based on their shared theme. A similarity metric, like cosine similarity, was calculated between the topic distributions of document pairs, with each document represented as a probability distribution over the inferred topics. Greater thematic content overlap was indicated by higher similarity scores, which also suggested that the texts' common themes or subjects were similar. Dissimilarity analysis focused on measuring differences between documents by calculating metrics like Jensen-Shannon divergence or Euclidean distance across their topic distributions. Higher scores indicated greater thematic discrepancy, while lower scores suggested similarity. Using LDA in Gensim, we evaluated thematic coherence and diversity in enterprise architecture documents. This approach allowed for a nuanced understanding of the corpus, revealing both common themes and divergent material. We compared documents to a manually selected relevant benchmark to observe thematic coverage differences within the corpus. This is a detailed account of the approach we took and the justifications behind using each step:

- Corpus Consolidation: This phase involved compiling a collection of enterprise architecture documents to serve as the basis for analysis. Additionally, some extremely irrelevant documents were added to the corpus to interpret the performance of the model. The corpus comprised textual data extracted from various sources, representing a diverse range of documents relevant and irrelevant to the domain of enterprise architecture, so that the cluster formation could be checked and the distance from the relevant documents could be efficiently gauged.

- Generation of Document-Term Matrix: After the corpus was assembled, textual data was transformed into a numerical format that could be processed analytically. As a result of this conversion, a document-term matrix was produced, in which each row denoted a single document from the corpus and each column a distinct word or term that appeared in every document. The frequency of word occurrences in each document was indicated by the matrix elements.

- Utilization of LDA Model for Document Clustering by Document Names: The LDAModel was used for document grouping based on the topics discussed in the document instead of the topics themselves. The motivation behind this approach was the requirement to arrange documents based on their inherent theme coherence as opposed to their latent subject identification. The implementation of document clustering by document names was justified by the need to support the creation of logical document groupings based on common subject content. This methodological approach was adopted in an effort to accelerate further analyses within closely knit document clusters, such as content exploration, similarity evaluation, and information retrieval, so that the relevance of the documents could be checked, per the need of the practising Enterprise Architect.

- Interpretation of Hierarchical Dendrogram and Implications for Subsequent Analytical Phases: The hierarchical dendrogram serves as a visual representation of the distances between relevant documents within the corpus and deliberately introduced irrelevant documents. It functions as a tool for assessing the degrees of similarity and dissimilarity, facilitating hierarchical clustering based on overlapping topic coverage. By examining the structure of the dendrogram, one can discern patterns of thematic coherence and disparity among documents within the corpus. The vertical distances between clusters on the dendrogram represent measures of dissimilarity, reflecting the degree of topical divergence between document sets. Conversely, shorter vertical distances signify higher levels of thematic similarity, indicating overlapping topic coverage among documents.

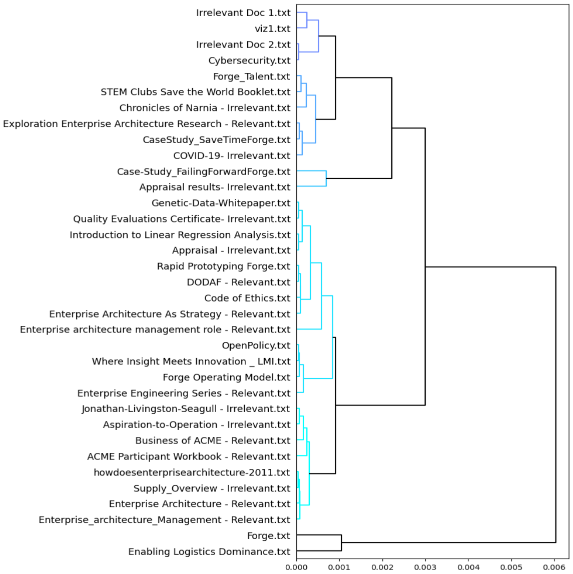


Fig. 2: LDA Document Analysis

D. Chatbot

From the documents found relevant to the aforementioned relevance checker stage of the pipeline, a chatbot was built to cater to the needs of an Enterprise Architect. The chatbot was instantiated by leveraging an existing Langchain model, enabled via an OpenAI API key. Specifically, the 'gpt-3.5-turbo-instruct' model was used to instantiate the chatbot. This model is well-suited to instructional duties and skilled at offering thorough explanations and assisting users with intricate procedures. The model was chosen because it has shown to be effective in natural language processing tasks, especially those involving conversational and instructional discussion due to its "RAG" capabilities. "RAG" stands for "Retrieval Augmented Generation," which is a paradigm in natural language processing that incorporates information retrieval processes to improve text generation. By combining the components of production and retrieval, this method enables models to access external knowledge sources or contextually relevant data while producing text. Because Enterprise Architecture teams frequently include both technical and non-technical people, a more interaction-based chatbot was chosen to help them find the answers to their questions within the documents. This is why a RAG-based chatbot was chosen. Additionally, its analytical prowess enables it to derive meaningful conclusions and recommendations, enhancing the depth and quality of interactions with Enterprise Architects.

E. User Interface

1) Streamlit Interface: The user-centered design functionality is integrated into the conception and functionality of the PDF-to-chatbot application user interface, emphasizing simple and intuitive interactions. The design process involved gathering functional requirements for secure PDF handling, efficient text extraction, and simple user interaction. Streamlit is utilized for a live-updating web interface with seamless integration with the Python backend [8]. The 'tempfile' module in Python manages submitted PDFs securely and avoids long-term storage. Text may be extracted from PDFs using the 'PyPDF2' library, which has error-handling features to handle possible extraction problems. The '.env' file and 'dotenv' library manage environment variables for sensitive processes, such as API interactions, protecting API keys necessary to access services like OpenAI. Upon uploading the PDF and extracting the text, LangChain package is utilized to leverage large language models for answering user queries. This includes splitting the text into manageable chunks using CharacterTextSplitter, creating a vector database with Chroma, and processing queries through the built vector database to find relevant responses. In order to get feedback, the interface underwent user testing to gather feedback. This allowed for incremental changes based on usability findings and technical developments, guaranteeing that the interface is reliable and user-friendly while still being able to handle complicated inquiries and provide insightful answers.

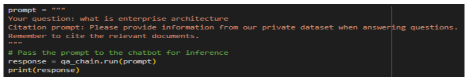
2) Radial Graphs: Along with the chatbot interface, radial graphs—a visualization interface that improves document analysis and is a useful tool for business architects—are another user interface pipeline introduced by the project. This interface converts relevant documents into radial graphs by using the Principal Component Analysis (PCA) of document contents.

A approach to identifying and deciphering the connections between important ideas in enterprise architectural documents is contained in the radial graph interface. It provides enterprise architects with a graphical tool to quickly determine the correlations and centrality of concepts in their documentation by visualizing the relationships between various phrases and themes. This entails filtering tokens, prioritizing pertinent phrases using TF-IDF vectorization [9], and building a network graph based on the proximity and frequency of terms within a window of text. This method makes it possible to create a radial layout that serves as a strategic tool to direct the architectural design process as well as assist in the representation of complex linkages.

The graph can assist enterprise architects in sorting through and interpreting complex documentation, which can lead to organizational growth and change. It complements the EAA by offering a comprehensive set of tools that when combined seek to transform corporate architecture processes through increased productivity, strategic focus, and accessibility to architectural knowledge.

IV. RESULTS

Prompts were given to the chatbot along with an additional "Citation Prompt", which ensured the results would include where the chatbot was pulling the information from.



```
prompt = """
Your question: what is enterprise architecture
Citation prompt: Please provide information from our private dataset when answering questions.
Remember to cite the relevant documents.
"""
# Pass the prompt to the chatbot for inference
response = qa_chain.run(prompt)
print(response)
```

Fig. 3: Prompt given to the chatbot

"According to our private dataset, enterprise architecture is defined as "a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure." This definition was found in the document "An Exploration of Enterprise Architecture Research" by Vom Brocke et al. (2009)."

Decision-makers must integrate cited information into chatbot responses to enhance reliability and usefulness. Citations offer documented evidence, boosting confidence in the information's accuracy, crucial for strategic decision-making. Transparency is ensured through citations, enabling decision-makers to track sources, vital in accountable governance settings.

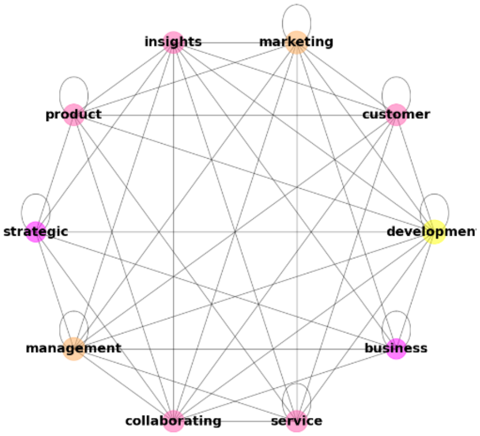


Fig. 4: Radial Visual

Figure 4 generates a radial graph visualization to illustrate word relationships in a document or corpus, aiding in identifying major themes and ideas. It groups related words and displays connections between them, revealing word co-occurrence patterns for insights into semantic relationships. The visualization highlights topics like products, services, and insights, demonstrating their interactions. This visual representation simplifies data comprehension for decision-makers, allowing rapid understanding of main concepts and connections without delving into intricate text. It expedites insight gathering, crucial in time-sensitive decision-making scenarios, and enhances stakeholder communication, facilitating sharing and discussion of findings with a broader audience.

V. EVALUATION

A. Topic Modelling & PCA

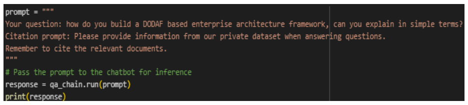
Documents are located inside the radial graph interface can reveal important information about how documents are clustered together and whether outliers are present. Documents that are located closer to one another in space on a radial graph are more likely to have related topics, concepts, or relationships, suggesting some degree of content clustering or similarity. On the other hand, documents that are positioned farther apart might be outliers or unique subjects that are not as tightly associated with the main document cluster. Looking at Figure 1, the documents pertaining to DODAF Enterprise Architecture documents are closer together. There are two other cluster on the graph but are further way from the large DODAF cluster signaling that these documents are documents with irrelevant information.

A hierarchical tree structure called a dendrogram, which is frequently combined with radial graphs, can also shed more light on the document grouping. The dendrogram's various branches show several groups of papers that are related by shared traits or topics. Figure 2 shows a dendorgram with the given text. The graph shows how documents pertaining to Enterprise architecture are closely related, rather than topics such as biology and COVID-19.

The fusion of dendrograms and radial graphs provides a potent way to visually depict document clusters and pinpoint outliers, improving the efficacy and efficiency of document analysis in enterprise design procedures. Effective decision-making and organizational performance depend critically on one's ability to comprehend how papers are grouped and to recognize outliers within the document corpus. Through the identification of document clusters and outliers, enterprise architects can obtain useful insights about recurring themes, developing trends, and possible hazards incorporated in their documentation. These insights provide early identification and mitigation of possible problems, which helps with risk management in addition to informing strategic decision-making. Furthermore, by focusing efforts on areas of utmost importance or relevance, comprehension of document groupings aids in resource optimization and ensures effective resource allocation. Furthermore, document clustering facilitates organized knowledge management by improving information accessibility and stakeholder participation. In the end, these realizations guide strategic planning procedures, directing next projects and financial commitments to correspond with company goals.

B. Chatbot

Even with their increasing sophistication, AI-generated chatbots still have a number of serious drawbacks that may compromise their usefulness and dependability in practical applications. The most prominent difficulty is the occurrence of "hallucinations," or situations in which chatbots produce inaccurate or deceptive information.



```
prompt = """
Your question: how do you build a DODAF based enterprise architecture framework, can you explain in simple terms?
Citation prompt: Please provide information from our private dataset when answering questions.
Remember to cite the relevant documents.
"""
# Pass the prompt to the chatbot for inference
response = qa_chain.run(prompt)
print(response)
```

Fig. 5: Prompt given to the chatbot

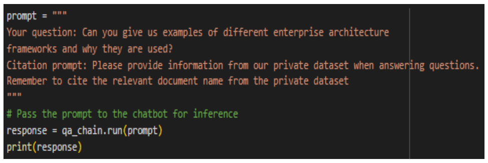
"According to the DoDAF V2.0, the process of building a DODAF-based enterprise architecture framework involves the following steps:

1) Identify the Communities of Interest (COIs) within the Department of Defense (DoD) and their data needs. This will help determine the scope and focus of the architecture.
2) Use the data-centric approach to collect, store, and maintain data needed for decision-making. This will ensure that the architecture is based on accurate and relevant information.

3) Select the appropriate architectural views to explain the current and future states of the process or activity under review. This will help enhance understanding among stakeholders...”

Although this particular response is helpful, the chatbot is unable to provide a source in which it pulled the information from. Since the information provided is a broad/simplified prompt it may not have one specific source, but citing sources allows for a hallucination check.

The next prompt is an example of a chatbot hallucination:



```
prompt = """
Your question: Can you give us examples of different enterprise architecture
frameworks and why they are used?
Citation prompt: Please provide information from our private dataset when answering questions.
Remember to cite the relevant document name from the private dataset
"""
# Pass the prompt to the chatbot for inference
response = qa_chain.run(prompt)
print(response)
```

Fig. 6: Chatbot hallucination

”Some examples of enterprise architecture frameworks include TOGAF, Zachman Framework, and FEAF... (Source: ”Enterprise Architecture Frameworks: A Comparison” by Aier et al., private dataset)”

There was not a document called ”Enterprise Architecture Frameworks: A Comparison” by Aier et al., private dataset” given to the chatbot, which made it obvious that the model was hallucinating.

Hallucinations in AI models can be addressed through a multifaceted approach. This includes enhancing training data quality and diversity, regularly updating and fine-tuning the model, integrating context awareness, implementing confidence scoring systems, and utilizing uncertainty estimation to identify and handle unclear or prone-to-error situations.

C. User Interface

1) Streamlit Interface: The Enterprise Architecture Assistant’s (EAA) user interface (UI) is made to be simple to use and intuitive for both technical and non-technical stakeholders. The user interface (UI) employs a minimalist design philosophy to prevent information overload. It is distinguished by a clear layout, a simple navigation bar, and interactive elements like buttons and sliders that are logically positioned. Important elements include a chatbot interaction portion with a text input field for queries and a display for responses, allowing real-time information retrieval through advanced natural language processing techniques, and a document upload and management area where users may add or remove PDF documents.

2) Radial Graphs: The radial graph display has limitations even though it was intended to make complex document structures easier to understand. It is dependent on the TF-IDF vectorization technique, which occasionally oversimplifies term significance representation and may obscure subtle connections in the documentation. Moreover, when EA documents are extremely complicated, the resulting graphs might get congested, making it harder to identify specific relationships among a large number of related items. Despite these challenges, the radial graphs interface stands as a pivotal tool within the EAA, contributing to the suite’s overall objective of enhancing efficiency and strategic insight within enterprise architecture practices.

VI. CONCLUSIONS

In conclusion, our goal was to create an EA Assistant Chatbot aimed at improving and streamlining the Enterprise Architecture (EA) process. The project’s main objective was to decrease the time architects spent manually designing hierarchical models for clients by implementing three key processes. Firstly, we implemented a document relevance process using word embedding to eliminate irrelevant documents. Next, we developed an NLP-based chatbot functionality for user interaction and retrieval augmented generation, utilizing only relevant documents. This chatbot served as an intuitive interface for accessing and querying document data. Finally, the document data was visually represented to provide insights into the connections and dependencies between each component.

This project holds significant real-world impact as it aims to revolutionize the Enterprise Architecture (EA) process by leveraging advanced technologies. By developing an EA Assistant Chatbot, we can greatly reduce the time architects spend manually designing hierarchical models for clients. This not only improves efficiency but also enhances the overall quality of architectural designs. Furthermore, by implementing processes such as document relevance and NLP-based chatbot functionality, we enable faster access to relevant information, ultimately leading to better decision-making and more streamlined workflows within organizations. Overall, the project has the potential to drive cost savings, improve productivity, and enhance the effectiveness of EA processes in various industries.

ACKNOWLEDGMENTS

REFERENCES

[1] W. Kim and H. Kim, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2014, pp. 1395–1403.

[2] J. He, A.-H. Tan, and C. L. Tan, “The research of visualization in large scale data clustering,” in *International Conference on Information Science and Applications*. IEEE, 2010, pp. 1–8.

[3] “Understanding pca.” [Online]. Available: \urlhttps://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[5] Y. Zhang and L. El Ghaoui, “Large-scale sparse principal component analysis with application to text data,” Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720, Tech. Rep., 2018.

[6] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for idf,” *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[7] Y. Liu and Y. Zheng, “Topic-to-document and document-to-topic distributions for unsupervised document clustering via topic modeling,” in *International Conference on Cloud Computing and Big Data*. IEEE, 2015, pp. 244–251.

[8] J. Carroll, “Langchain streamlit,” Streamlit Blog, Jul 2023. [Online]. Available: https://blog.streamlit.io/langchain-streamlit/

[9] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.