CHAT-ACADEMIC: Chat with your academic papers with LLM

School of Computer Science
University of Nottingham
Nottingham, UK

Hongye An
School of Computer Science
University of Nottingham
Nottingham, UK
psxah15@nottingham.ac.uk

***Abstract*—In the realm of academic paper retrieval, conventional methods involve the utilization of specialized tools such as Google Scholar, requiring researchers to invest considerable time navigating through an extensive array of scholarly articles to identify those aligning with specific criteria. This endeavor demands notable dedication and incurs time-related expenditures. In response to these challenges, this paper introduces a novel tool named CHAT-ACADEMIC, designed to enhance the academic paper retrieval and literature review process through the utilization of large language models (LLMs) [1] and vector database technology.**

CHAT-ACADEMIC enhances the efficiency of users in retrieving relevant literature. It addresses the limitations of traditional paper search engines, such as Google Scholar, by employing Text Embedding technology. The system offers an interactive approach based on natural language dialogue, allowing users to search for papers based on the semantics of titles or abstracts. Simultaneously, it visualizes the embedding space by reducing high-dimensional vectors, aiding users in rapidly identifying semantically similar articles.

***Index Terms*—LLMs, text embedding, vector database, literature review, data visualization**

I. Introduction

In recent years, the proliferation of vast digital repositories of academic papers has posed a dual challenge for researchers and scholars alike—efficiently retrieving pertinent information from this expansive pool and comprehensively visualizing the intricate relationships within scholarly literature. Traditional information retrieval methods often fall short in capturing the nuanced connections between academic papers, thereby hindering the seamless extraction of relevant insights. [2] This paper proposes a pioneering solution in the form of a Retrieval Augmented Generation (RAG) application [3], which leverages the capabilities of Large Language Models (LLMs), vector databases, and Relationship Diagram Visualization to address the aforementioned challenges.

The RAG framework harnesses the power of state-of-the-art language models (such as GPT-3.5 and text-embedding-ada-002 by OpenAI [4]), to facilitate dynamic and contextually aware conversations for academic paper retrieval. By combining advanced natural language processing with a rich vector database, our application enables users to articulate complex queries in a conversational manner, fostering a more intuitive and interactive information retrieval experience. This marks a departure from traditional keyword-based approaches, affording users the ability to express nuanced search criteria and extract more precise information from the vast academic corpus.

In conclusion, the fusion of retrieval augmented generation techniques with cutting-edge visualization methodologies holds promise in overcoming the limitations of conventional approaches to academic paper exploration. The RAG application, as detailed in this paper, represents a significant stride towards enhancing the efficiency and depth of scholarly information retrieval, ushering in a new era of interactive and context-aware exploration within the academic domain.

II. Related Work

A. Text Embedding

Text Embedding [5], a technique widely used in natural language processing, facilitates semantic similarity searches by representing words or documents as dense vectors in a continuous vector space. This method leverages the distributional hypothesis, which posits that words appearing in similar contexts tend to have similar meanings. By training on large corpora of text data, Text Embedding models learn to capture semantic relationships between words and phrases. Consequently, words with similar meanings are mapped to nearby points in the vector space. Thus, Text Embedding enables semantic similarity searches by measuring distances or similarities between vectors, allowing for effective retrieval of documents based on their semantic content.

Text embedding serves the purpose of converting textual data into high-dimensional vectors, effectively capturing semantic relationships. To accomplish this, the research leverages pre-trained embedding models, specifically employing well-established models such as Word2Vec [6] or [7] embeddings. But ChatAcademi will use the model text-embedding-ada-020 by OpenAI, this model was renowned for its ability to generate contextually rich embeddings by leveraging advanced attention mechanisms and transformer architectures. It is noteworthy that, in contrast to traditional methods, this paper utilizes the application programming interface (API) provided by OpenAI for text embedding. The implementation of this approach yields a 1536-dimensional vector representation. The vector will be stored in the vector database, it can store vectors along with other data items. The similarity between texts can

be judged by the spatial distance (Euclidean Distance, Cosine Distance, Dot Product Distance etc.) between vectors.

B. Vector Database

A Vector Database [8] serves as a specialized storage and retrieval system tailored for managing high-dimensional embeddings that represent entities or documents. Unlike traditional databases, which primarily handle structured or unstructured textual data, Vector Databases are optimized for storing and querying dense embeddings efficiently. One key distinction between Vector Databases and traditional databases lies in their data representation and retrieval methods. While traditional databases rely on indexing and querying based on structured data fields or text content, Vector Databases employ techniques tailored for high-dimensional vector spaces. These databases often utilize specialized algorithms for approximate nearest neighbor search, which efficiently locate vectors that are close to a given query vector in the vector space.

By leveraging Vector Databases, CHAT-ACADEMIC can perform semantic similarity searches at scale, efficiently retrieving documents or entities that are semantically similar to a given query. This capability is particularly valuable in applications where understanding semantic relationships between data points is crucial, such as content recommendation, similarity-based search, and clustering analysis [9]. Fig.1 shows the process of text semantic similarity retrieval using vector database and embeddings. Overall, Vector Databases play a vital role in enabling advanced semantic search capabilities and supporting complex data analysis tasks in modern information systems.

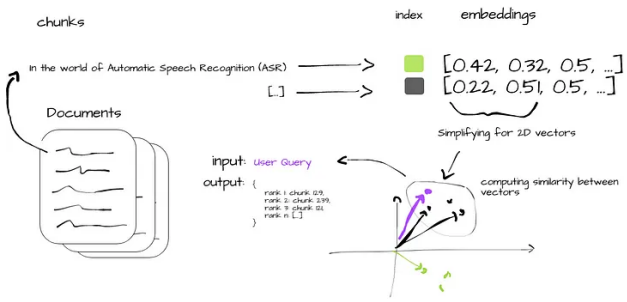

Fig. 1. The flow of semantic similarity search. [10] (image by Luís Roque) (1) Collect raw data. (2) Cut into chunks. (3) Text transform to embeddings. (4) Database indexing. (5) Vector distance search.

CHAT-ACADEMIC will utilize ChromaDB as its vector database solution. ChromaDB is an open-source, lightweight, and easily deployable vector database system. It efficiently supports the storage and retrieval of embeddings. Additionally, ChromaDB integrates third-party products such as LangChain and OpenAI API to facilitate the rapid and efficient development of LLM applications.

C. Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG) is a powerful methodology for natural language processing and generation tasks, incorporating aspects of both retrieval from databases and generation using language models. The RAG model essentially combines the strengths of language models like GPT with information retrieved from a large corpus or database to generate more informed, accurate, and detailed outputs. The CHAT-ACADEMIC adopts the design principles of RAG, integrating components such as LLMs, vector databases, and embedding models to develop a RAG application [19]. This innovative approach revolutionizes the workflow of literature search processes, enhancing both efficiency and user experience for individuals engaged in scholarly research endeavors.

In the field of literature review, CHAT-ACADEMIC uses RAG approach to have the following obvious advantages:

1. **Content Coverage**: In the process of literature review, it is essential to comprehensively gather and analyze relevant research literature in the respective field. RAG, through retrieving pertinent information from large-scale databases, aids researchers in discovering a broader spectrum of relevant studies and materials, thereby enhancing the comprehensiveness and depth of literature reviews.

2. **Enhanced Efficiency**: Traditional literature review procedures demand substantial time investments from researchers in literature retrieval and reading. Leveraging RAG technology automates a portion of the retrieval and preliminary analysis tasks, swiftly extracting key information from a vast corpus of literature, consequently significantly improving work efficiency.

3. **Customization and Personalization**: Tailored to different research domains and literature review objectives, RAG systems can be customized and personalized through adjustments in the papers data sources or databases for retrieval and optimization of generation models. Such adaptations cater to diverse users and requirements, offering more precise and high-quality services.

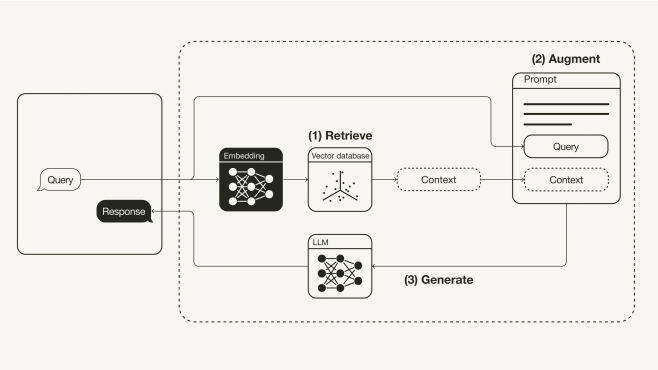

Fig. 2. The basic pipeline of RAG application [11] (image by Leonie Monigatti) (1) Retrieve. (2) Argument. (3) Generate.

The mainly RAG pipeline includes the three key steps (as shown in Fig.2), the following content will introduce how CHAT-ACADEMIC will do in these steps:

1. **Retrieve**: This is the first and crucial phase where the system searches for and retrieves relevant information from an external database or document set that matches the user input query or prompt. This retrieval process is not about finding

exact answers but rather fetching documents that contain potentially useful context or information related to the query. CHAT-ACADEMIC will embed the content entered by the user with the same embedding model, and then search the vector database for content with similar semantics.

2. **Argument**: In this phase, the retrieved documents are processed and fused into the context used by the model for generating the response. This step involves integrating the relevant information from the retrieved documents with the initial prompt or query to enrich the context. Usually, we could customize the prompt template, insert the searched documents into the template, and use them to enhance the interaction with LLM in subsequent steps [20]. For instance, in scenarios akin to those handled by LangChain [12], the default template for prompts is as follows:

Use the provided context to formulate a response to the following question. If uncertain, it is preferable to acknowledge a lack of knowledge rather than generating a speculative response.

{*context*}

Question: {*question*}

In this prompt template example, the content within the "context" section corresponds to the documents retrieved in the **Step1** (Retrieve) that exhibit relevance. The "question" represents the user's original inquiry. By designing specific prompt templates such like this, CHAT-ACADEMIC can realize the enhancement and responsiveness of the literature review system.

3. **Generate**: The final phase, CHAT-ACADEMIC leverages a LLM (GPT model here) to produce a coherent and contextually relevant response or text output. This generation is informed by both the original input and the augmented context provided by the retrieved documents. The LLM effectively combine information from its literature database and the augmented external knowledge to create responses that are significantly more informed, accurate, and diverse. The generative step is where the creativity of LLM shows, as it can produce novel sentences and paragraphs that were not explicitly found in the input data or the retrieved documents, enriching the final output with insights drawn from a broad knowledge base.

D. Chaining LLM Prompts [13]

The dialogue module stands as the foremost and prominent constituent of the current system. However, a pending issue within this system is the effective handling of diverse user inputs. Assuming that each user input is consistently formulated as a request for a specific type of article, such as *"Please help me search for some papers related to data visualization"* The system can indeed be designed in accordance with the methodology expounded in the preceding RAG pipeline section. However, if the content of user input is framed as *"Please help me summarize the general content of the XXX paper you mentioned"* or *"Please tell me about the research methods in the XXX paper"*, among some user inputs where the user is not necessarily seeking a query of papers related to a specific topic in the database, but rather engaging in other actions—the system ideally should generate appropriate responses. Although this could be achieved through the design of prompt structures, such an approach risks introducing complexity and intricacy into the prompt design, potentially triggering hallucination issues [18] within the Language Model (LLM).

To address this issue, the current project intends to incorporate LangChain [12] and draw inspiration from its design principles in the subsequent implementation phase for the development of the dialogue processing system. The design philosophy of LangChain, when addressing intricate challenges of this nature, introduces the concept of chaining together steps within the Language Model (LLM), where the output of one step serves as the input for the subsequent step, thereby aggregating the benefits of each step (as depicted in Fig.3). This necessitates developers to decompose the task into distinct, specific subtasks, subsequently aggregating the results of these subtasks to create a more robust functionality and achieve more precise responses to user instructions within the RAG application.

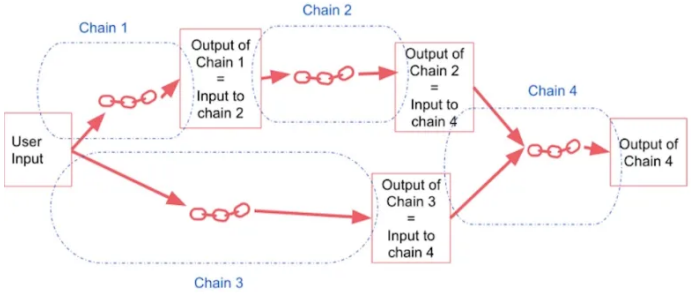

Fig. 3. The "chain" model introduced by Langchain. [14] (image by Babina Banjara)

Another issue pertains to the management of context in the RAG application. In this context, we aim for our project system to "remember" the dialogue history between the user and the system. However, since the LLM API system does not inherently assist in recording contextual content each time the system makes a request to the LLM interface, it necessitates our own design and management of the context of the user-system dialogue. We may also draw inspiration from LangChain's design philosophy regarding the "context manager" The specific approach involves sending all content of the current session dialogue to the LLM interface during each interaction, allowing the LLM to respond based on the entire conversation history. However, in practical implementation, given that many LLMs impose constraints on the token count for user requests, our system must consider appropriate truncation of content when the contextual content in a session exceeds the limit. Alternatively, we may utilize the LLM API for concise summarization, thereby enhancing user immersion in the "conversation" with our system.

E. Academic Literature Visualization

Information visualization is an important part of the CHAT-ACADEMIC system. Karin's research [15] points out that visualization can have a significant impact on people's decision-making. In this pivotal section dedicated to visualization, the primary objective here is to present a visually intuitive and informative display of the inter-document similarity, thereby augmenting the user's ability to conduct scholarly literature retrieval with enhanced precision and insight. Since the vector database store all the academic landscape embeddings, wherein each academic paper is encapsulated in a high-dimensional vector representation. This paper employ the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm [16], a sophisticated dimensionality reduction technique that excels in preserving local structures within high-dimensional datasets. t-SNE transforms complex vector embeddings into a lower dimensional space (2-D or 3-D for visualization) and perform clustering, effectively capturing the nuanced relationships between academic papers. By utilizing t-SNE, this paper aim to create a visually compelling representation that accurately reflects the inherent semantic connections among scholarly works.

The figure below (Fig.4) is an example of 2-D visualization of vectors using python plot library and the t-SNE algorithm. It can be seen that each node represents an actual document in the database, and the nodes are connected through line segments. The same color represents a certain degree of similarity between these vectors in the space. CHAT-ACADEMIC will also introduce similar visualizations within the user interface to display papers with similar semantics.

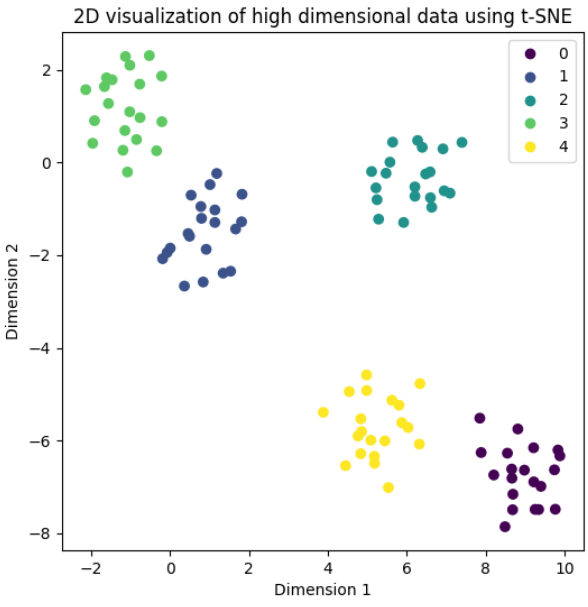

Fig. 4. An example of using the t-SNE algorithm and python plot library to project high-dimensional data into a 2-D space and distinguishing each type of color.

III. Formative Study

In order to enhance the design of the literature retrieval tool, a user case study was conducted. Two users were selected for interviews, one being a Master's student in computer science, and the other a PhD candidate specializing in geographic information science. Both individuals had previous experience in conducting literature searches and organizing information. The interviews were conducted primarily through online video conferences, with a duration ranging from 20 to 30 minutes.

The interview format followed a conversational question-and-answer approach. Specific questions may have varied based on each individual's distinct academic background, but the inquiries primarily focused on the following aspects:

1. The academic backgrounds and professional habits of users, specifically encompassing the purposes, frequency, and favored tools employed in academic paper retrieval.

2. The evaluation and user experience of existing academic paper retrieval tools, primarily centered around the functionality of the tools, including dimensions such as strengths, weaknesses, and areas for improvement.

3. The overall experience of the current academic paper search workflow, inquiring about users' foremost challenges and pain points within the existing workflow.

4. Regarding the existing academic paper retrieval tools, soliciting feedback on the UI design, interactions, and visualization of results, as well as understanding how users wish for academic papers to be presented.

A. The purpose of the user's literature review

Based on the summarized results of user interviews, both interviewed users exhibit a need for academic research and the composition of scholarly papers. The following discusses the user's purpose when conducting a literature search.

In the initial stage, users typically seek to query academic papers relevant to their research domains. By searching for scholarly articles within their respective fields, users gain access to literature reviews summarizing existing research, thereby comprehending the current state and theoretical foundations of their fields. This aids in constructing a research framework, ensuring a comprehensive understanding of the background and theoretical underpinnings of the research question, and providing reference perspectives for research design.

In the second phase of academic paper composition, users require support in the form of specific data and evidence. During this stage, engaging in academic paper searches proves beneficial in acquiring additional empirical data and supporting materials. Users can explore the latest research findings to bolster their data analysis and research conclusions, enhancing the credibility of their papers. Furthermore, these identified papers can be utilized as references, contributing to the substantiation of the users' scholarly work.

B. User feedback

In addressing the current workflow, insights were derived from interviews with participants who provided feedback

on challenges encountered when utilizing certain literature retrieval tools, such as Google Scholar.

Primarily, a notable concern revolves around the prevalent use of keyword-based searches in contemporary literature retrieval tools. The inherent limitation of keyword-based searches lies in the reliance on users to supply keywords for matching, which can prove challenging as users may struggle to precisely articulate their needs, resulting in less accurate search outcomes. Ambiguities inherent in certain keywords further compound the issue, making it difficult for search engines to discern users' exact intentions.

Another significant issue pertains to the excessive and disorganized nature of the information retrieved through searches. Search results may encompass a vast array of both relevant and irrelevant literature, rendering it challenging for users to sift through and discern genuinely valuable information. Consequently, users expend substantial time reviewing the content of queried literature, significantly augmenting the complexity of the retrieval process.

C. Design goals

In summary, based on the interviews conducted with the respondents, we have formulated the following design objectives for the project:

1. **Innovation**: The cornerstone of CHAT-ACADEMIC is to pioneer a novel approach to literature retrieval. The essence of this innovation lies in introducing a conversation-based interaction method, setting the system apart from traditional query-based search engines. This method hinges on understanding and engaging with users through contextual dialog, allowing the system to grasp the nuances of user queries beyond mere keywords. It aims to mimic a natural, human-like conversation, adapting to the user's needs and questions in real time. To accomplish this, our system will leverage cutting-edge LLM techniques to interpret, respond, and learn from the context of the conversation.

2. **Functionality**: The functionality objectives of CHAT-ACADEMIC encompass the following aspects: 1. To transcend the traditional keyword search paradigm by facilitating semantic searches of related literature. 2. To chronicle the history of users' literature search dialogues. 3. To offer a web-based user interface. 4. To visually present the results of literature searches.

3. **Visualization**: A vital component of enhancing user experience in literature search is visualization. CHAT-ACADEMIC intends to incorporate a sophisticated visualization interface, particularly when the user's input pertains to a request for literature search. Through this interface, the semantic relationships among the search results will be visually embodied in the form of a 2-D network node graph. Each node in this graph represents a piece of literature, nodes with the same color represent similar semantics. Meanwhile, the paper retrieved results obtained from the search are displayed in the form of a list view, and interactive effects are added to the relationship network diagram and academic paper list view. This visual representation will enable users to intuitively navigate through the corpus of literature, uncover patterns and relationships they might not have noticed, and ultimately, facilitate a deeper understanding of the subject matter at hand.

IV. CHAT-ACADEMIC

We presented CHAT-ACADEMIC, this system aims to use LLM technology to empower the literature search or review step for academic researches

A. Data collection

For the data component of this project, also known as the literature database available for user search queries, This project have opted for the data visualization literature database from VITALITY [2].

VITALITY is another literature search tool designed by Arpit et al [2]. In order to enrich the dataset, Arpit et al. developed a web scraping module to extract academic papers relevant to the field of data visualization from the DBLP dataset [17]. This extraction encompassed various attributes such as the publisher of each article (e.g., IEEE Xplore, ACM Digital Library), abstracts, keywords, and citation counts. Subsequently, the dataset underwent filtering, removing certain authors and papers with empty abstracts, as well as titles and abstracts of either very short or very long length. The final result is a scholarly database within the visualization domain, containing approximately 59,000+ academic papers.

B. UI Design

The system UI design shown in Fig.5. As illustrated in the figure, the UI is divided into three distinct regions: the left section comprises functionalities and a history of dialogues, the central area is designated for displaying conversation text, and the right segment encompasses additional functionalities.

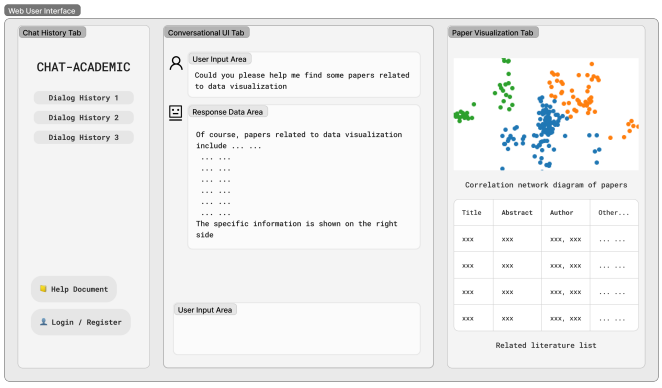

Fig. 5. User interface design diagram

1. **Chat History Tab**: As depicted in the diagram, on the left side is the "Chat History Tab", which displays a list of the user's dialog history, as well as the Help Document, and the functionalities for Login/Register. Upon the completion of each dialogue, all content exchanged between the user and this system in the current session will be stored. Consequently, this feature allows users to conveniently and effortlessly review

historical information, facilitating the recollection of past dialogues.

2. **Conversational UI Tab**: The middle part of the design diagram representing its most critical element. At the upper portion of this segment, the dialogue content that is reciprocally exchanged between the user and the system is displayed, establishing an interactive communication platform. Directly below this area is an input box designated for user contributions, facilitating the entry of relevant information or queries. This is an important part of CHAT-ACADEMIC's conversational interaction.

3. **Paper Visualization Tab**: Diverging from typical conversational agents that predominantly feature a chat interface, this project, being a dedicated RAG application focused on assisting users in enhancing academic literature searches, incorporates a distinctive design. The upper section of the area mentioned above is the embedding dimensionality reduction visualization chart implemented using the t-SNE algorithm, as previously discussed. This part can display the searched papers with similar semantics. Additionally, this section offers interactive operations allowing users to view the related literature. The lower section, on the other hand, employs a conventional list format to present a comprehensive list of literature obtained through the retrieval process, including key information such as article titles, abstracts, authors, etc.

V. Future Work

A. System Implementation

The current progress of this project includes the completion of the UI design diagrams and the interaction process design. However, the specific implementation is still in its prototype phase. The subsequent goal is to fully implement this system, providing a web-based UI interface. The backend service portion of this system will utilize a server-side framework based on the Python programming language, as Python can integrate seamlessly with LangChain and LLM. The frontend interface will be designed according to the aforementioned design diagrams and will communicate with the backend system via the HTTP network protocol. Ultimately, this will present users with a complete web page application.

B. Testing

Upon completing the implement of CHAT-ACADEMIC, it is essential to conduct systematic testing of the system, the testing contains the following key aspects:

1. **Functional Testing**: This phase involves verifying the implementation of each functional requirement. For instance, it assesses whether relevant academic papers can be retrieved through keyword searches and confirms if text embedding and vector database functionalities operate as intended to provide retrieval results for pertinent papers.

2. **Accuracy Testing**: Accuracy testing entails validating the relevance of retrieved academic papers. This can be accomplished using pre-constructed benchmark test sets comprising various queries and their expected retrieval outcomes. By comparing the application's output with benchmark results, accuracy can be quantified (e.g., using metrics such as precision, recall, etc.).

3. **Usability Testing**: Given that the target users of CHAT-ACADEMIC are researchers and students engaged in literature review, the application's UI/UX design should be intuitive and user-friendly. Usability testing can be conducted through user testing sessions, allowing the target user group to evaluate the application's ease of use, intuitiveness of navigation flow, and comprehensibility of functionalities.

C. Evaluation

Upon completing the project implementation, another crucial step is derived from formative feedback. We plan to recruit 3 - 4 volunteer users to experience our system, with volunteer feedback serving as the evaluation criteria. The assessment will be conducted through the following approaches:

1. **User Orientation**: Providing detailed information to volunteers about the system's background and functionalities, including interface design, feature segmentation, and instructions on system utilization.

2. **User Behavioral Testing**: Presenting users with specific tasks and scenarios to observe how they utilize the tool to accomplish these tasks. By observing and recording user behavior, data will be collected regarding interface navigation, search behaviors, and response times. Users will be encouraged to provide real-time feedback during usage to obtain immediate perspectives.

3. **Data Analysis**: Extracting key qualitative information from user testing, such as user experience, pain points, and suggestions. Analyzing this feedback to determine which user issues the system effectively addresses in real-world applications, as well as identifying areas that require further optimization.

References

[1] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D. and Chi, E.H., 2022. Emergent abilities of large language models. arXiv preprint arXiv:2206.07682.

[2] Narechania, A., Karduni, A., Wesslen, R. and Wall, E., 2021. Vitality: Promoting serendipitous discovery of academic literature with transformers & visual analytics. IEEE Transactions on Visualization and Computer Graphics, 28(1), pp.486-496.

[3] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.T., Rocktäschel, T. and Riedel, S., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, pp.9459-9474.

[4] OpenAI, O. (2023) OpenAI platform, Models - OpenAI API. Available at: https://platform.openai.com/docs/models/gpt-3-5 (Accessed: 09 January 2024).

[5] Tang, J., Qu, M. and Mei, Q., 2015, August. Pte: Predictive text embedding through large-scale heterogeneous text networks. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1165-1174).

[6] Church, K.W., 2017. Word2Vec. Natural Language Engineering, 23(1), pp.155-162.

[7] Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y., 2019. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.

[8] Pan, J.J., Wang, J. and Li, G., 2023. Survey of Vector Database Management Systems. arXiv preprint arXiv:2310.14021.

[9] Azhir, E., Navimipour, N.J., Hosseinzadeh, M., Sharifi, A. and Darwesh, A., 2021. An automatic clustering technique for query plan recommendation. Information Sciences, 545, pp.620-632.

[10] Roque, L. (2023) Document-oriented agents: A journey with vector databases, LLMS, Langchain, FASTAPI, and Docker, Medium. Available at: https://towardsdatascience.com/document-oriented-agents-a-journey-with-vector-databases-llms-langchain-fastapi-and-docker-be0efcd229f4 (Accessed: 26 December 2023).

[11] Monigatti, L. (2023) Retrieval-augmented generation (RAG): From theory to Langchain Implementation, Medium. Available at: https://towardsdatascience.com/retrieval-augmented-generation-rag-from-theory-to-langchain-implementation-4e9bd5f6a4f2 (Accessed: 15 January 2024).

[12] Langchain, L. (no date) Langchain, LangChain. Available at: https://www.langchain.com/ (Accessed: 21 January 2024).

[13] Wu, T., Terry, M. and Cai, C.J., 2022, April. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In Proceedings of the 2022 CHI conference on human factors in computing systems (pp. 1-22).

[14] Banjara, B. (2023) A comprehensive guide to using chains in Langchain, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2023/10/a-comprehensive-guide-to-using-chains-in-langchain/ (Accessed: 22 January 2024).

[15] Eberhard, K., 2023. The effects of visualization on judgment and decision-making: a systematic literature review. Management Review Quarterly, 73(1), pp.167-214.

[16] Van der Maaten, L. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(11).

[17] Schloss Dagstuhl - Leibniz Center for Informatics, dblp (2024) Home, dblp. Available at: https://dblp.org/ (Accessed: 20 January 2024).

[18] Yao, J.Y., Ning, K.P., Liu, Z.H., Ning, M.N. and Yuan, L., 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. arXiv preprint arXiv:2310.01469.

[19] Jeong, C., 2023. A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture. arXiv preprint arXiv:2309.01105.

[20] Feng, Z., Feng, X., Zhao, D., Yang, M. and Qin, B., 2023. Retrieval-generation synergy augmented large language models. arXiv preprint arXiv:2310.05149.