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 pursuit of academic paper retrieval, researchers typically rely on specialized tools like Google Scholar, investing substantial time in navigating an extensive array of scholarly articles to identify those aligning with specific criteria. This undertaking demands notable dedication and incurs time-related expenditures. This paper introduces an innovative tool leveraging large language models (LLMs) and vector database technology to augment users in the retrieval of academic papers and literature review. The proposed tool aims to facilitate a swift and precise identification of pertinent scholarly materials, streamlining the research process for enhanced efficiency. And present the results in the form of a 3-D visual network graph to illustrate the similarity and correlation among various papers.**

***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. This paper proposes a pioneering solution in the form of a Retrieval Augmented Generation (RAG) application[2], 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[3]), 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. Methodology

A. Text Embedding In Vector Database

To address the semantic disparity between user queries and stored information, this study employs text embedding[13] and vector database techniques. The integration of text embedding techniques and vector databases has emerged as a pivotal approach in enhancing the efficiency and accuracy of information retrieval systems. Text embedding involves the transformation of textual information into continuous vector representations, capturing semantic relationships and contextual nuances. Concurrently, vector databases provide a structured and efficient means of storing and querying these high-dimensional representations.

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 BERT[7] embeddings. But this paper 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.

The amalgamation of advanced text embedding models, exemplified by the "text-embedding-ada-00," and the utilization of ChromaDB[8] as a vector database underscores the commitment to enhancing the precision and relevance of information retrieval within the scholarly domain. In the subsequent sections, this paper delve into the specifics of the RAG architecture, elucidating how the symbiotic relationship between text embedding and vector database technologies forms a foundational element of our innovative approach to academic paper retrieval and visualization.

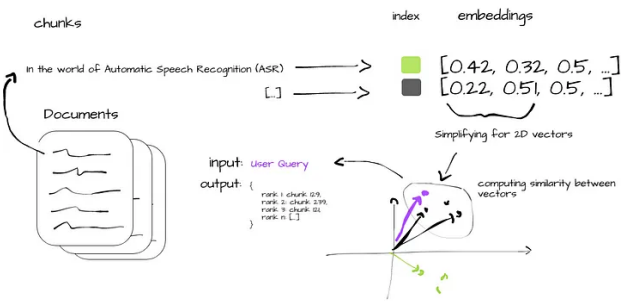

Fig. 1. The flow of semantic similarity search. [1] (image by Luís Roque) (1) Collect raw data. (2) Cut into chunks. (3) Text transform to embeddings. (4) Database indexing. (5) Vector distance search.

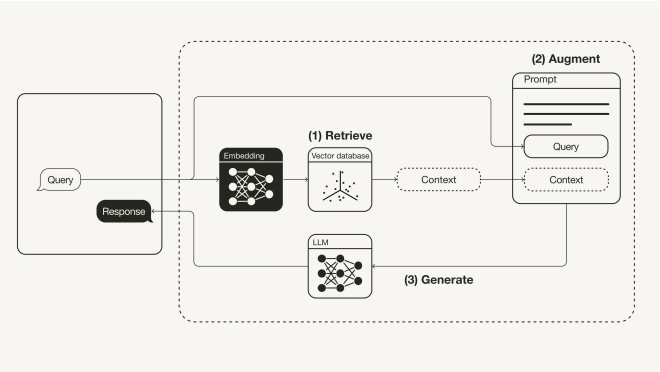

Fig. 2. The basic pipeline of RAG application (1) Retrieve. (2) Argument. (3) Generate.

B. RAG Application Pipeline

The Retrieval-Augmented Generation (RAG)[14] pipeline employed in this academic paper epitomizes a cutting-edge approach in the realm of conversational applications for academic paper retrieval and visualization. At its core, the RAG application harnesses the prowess of large language models, coupled with a meticulously curated vector database, to facilitate an intricate interplay between retrieval and generation processes. The intricate orchestration of these components ensures a synergistic fusion of advanced language understanding and content retrieval capabilities.

This mainly includes the following three key steps:

(1) **Retrieve**: The user's input information is embedded through the same model, and a search for similar text is conducted within the vector database. This approach is advantageous due to the token limitations inherent in many large model APIs. It is impractical to transmit the entire database of academic papers to the large model via API due to these constraints. Consequently, the utilization of a vector database for preliminary relevance searches becomes imperative, facilitating the retrieval of pertinent academic literature texts through correlation analysis.

(2) **Argument**: In this step, the primary objective involves invoking the interface of a Large Language Model (LLM) to enable it to respond to user queries based on the relevance documents obtained in the preceding retrieval step. For instance, in scenarios akin to those handled by LangChain, 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, the aim is to facilitate the augmentation of the LLM for improved responsiveness.

(3) **Generate**: In the context of this intricate procedural framework, the term "Generate" encapsulates the conclusive phase, signifying the culminating step where it becomes imperative to transmit the prompt derived from the antecedent stage, denoted as "argument," to the Large Language Model (LLM). In this critical juncture, the LLM assumes the responsibility of generating the essential response. This process involves a sophisticated synthesis of information inherent in the conveyed prompt, wherein the LLM employs its linguistic capabilities to formulate a coherent and contextually relevant answer, thus finalizing the comprehensive trajectory of the pipeline. The meticulous orchestration of this generation phase contributes significantly to the overall efficacy and proficiency of the conversational application, demonstrating the culmination of the retrieval-augmented generation (RAG) mechanism in the academic context.

C. Retrieval Result Visualization

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[12], a sophisticated dimensionality reduction technique that excels in preserving local structures within high-dimensional datasets. t-SNE transforms complex vector embeddings into a three-dimensional space, 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.

In the context of our academic paper embeddings, t-SNE will enable us to project these high-dimensional vectors into a three-dimensional space. The resulting visualization will showcase clusters of academic papers with similar thematic content, providing users with an insightful overview of the scholarly landscape.

To bring this conceptualization to life, this paper leverage the three.js framework, a powerful and flexible JavaScript library designed for 3D graphics. This framework empowers us to create an interactive and engaging visualization, allowing users to navigate the 3D representation of academic paper relationships. The integration of three.js aligns seamlessly with our commitment to delivering a user-centric and technologically advanced platform for academic literature exploration.

The figure below (Fig.3) is an example of 3-D visualization of vectors using three.js 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.

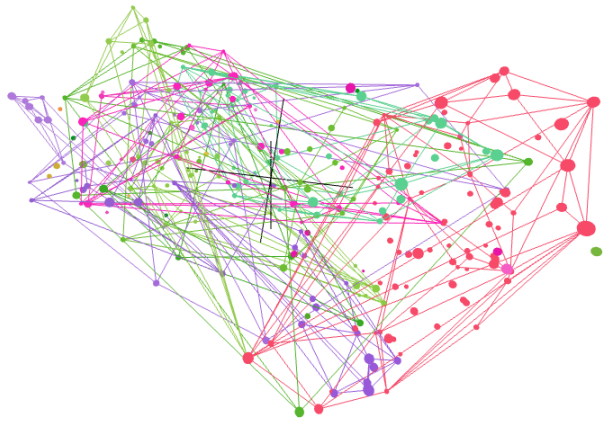

Fig. 3. Example of 3-D visualization of vectors using three.js and the t-SNE algorithm

III. User Case Studies

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 search

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 on current workflow of literature search

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**: Introduce a novel, conversation-based interaction method for literature retrieval, supporting the ability to answer user-specific questions based on contextual dialog.

2. **Functionality**: Employ advanced language models and text embedding techniques to realize literature retrieval functionality based on semantic relevance.

3. **Visualization**: In cases where the user's input involves a request for literature search, the interface will visually represent the semantic relationships among the search results in the form of a network node graph.

IV. Design

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[4].

VITALITY is another literature search tool designed by Arpit et al[4]. 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 [5]. 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.4. As illustrated in the figure, the UI is divided into three distinct regions: the left section comprises functionalities and a history log of dialogues, the central area is designated for displaying conversation text, and the right segment encompasses additional functionalities.

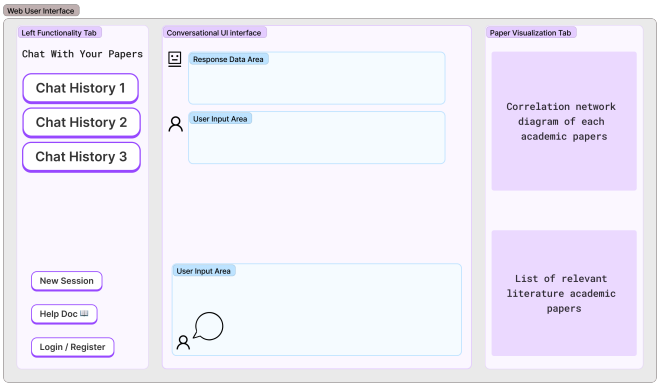

Fig. 4. User interface design diagram

1. **Left side functional area**: As depicted in the illustration, the upper section on the left displays a list of recent dialogues history undertaken by the user. Following the completion of each dialogue, a corresponding history record is documented in this section. Below this, options are available to initiate a new conversation, access help documentation, and utilize buttons for logging in or registering.

2. **Central dialogue area**: This segment constitutes the most pivotal and crucial component of the interface. The upper part of this section houses the dialogue content exchanged between the user and the system. Beneath this, an input box is provided for the user, allowing the entry of any pertinent content. The system, in turn, formulates responses based on the prevailing context of the ongoing dialogue.

3. **Right customization functionality area**: 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. In this context, the right segment of the interface is allocated for functionalities pertaining to the visualization of search results relevant to scholarly literature. One component within this region is dedicated to presenting a network graph depicting the textual relevance of academic literature retrieved through searches. Another subsection adopts a conventional list format to display the comprehensive list of literature obtained through the search process.

V. Future Plan

A. Dialog flow design

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[9] within the Language Model (LLM).

To address this issue, the current project intends to incorporate LangChain[10] 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.5). 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.

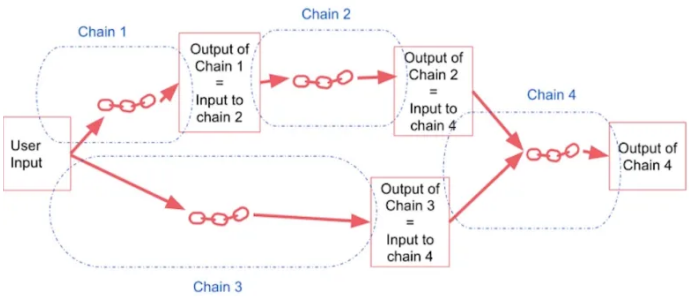

Fig. 5. The "chain" model introduced by Langchain.

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.

B. 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] 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).

[2] 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.

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

[4] 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.

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

[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] Șorecău, M. and Șorecău, E., 2023. An Alternative Application to CHATGPT that Uses Reliable Sources to Enhance the Learning Process. In International conference KNOWLEDGE-BASED ORGANIZATION (Vol. 29, No. 3, pp. 113-119).

[9] 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.

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

[11] 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).

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

[13] 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).

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