Journal of Theoretical and Applied Information Technology

30th November 2023. Vol.101. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BUILDING A RETRIEVAL-AUGMENTED GENERATION SYSTEM FOR ENHANCED STUDENT LEARNING: CASE STUDY AT PRIVATE UNIVERSITY

M BAGASKORO TRIWICAKSANA S[1] , TANTY OKTAVIA[2]

[1,2]Information System Management Department, BINUS Graduate Program – Master of Information System Management, Bina Nusantara University, Jakarta, Indonesia 11480

E-mail: [1]m.triwicaksana@binus.ac.id, [2]toktavia@binus.edu

ABSTRACT

This research conducted at Private University investigates the development and implementation of a Retrieval-Augmented Generation (RAG) system for enhanced student learning. The RAG system is a blend of retrieval-based and generative models using ChatGPT, aiming to address the challenges students face in accessing and understanding digital literature, mainly due to language barriers and passive reading methods. The RAG prototype was successfully created and assessed through black box testing and usability testing among students at Private University. Findings show that the RAG system significantly enhances interactive learning by providing contextually relevant answers. The system is highly functional and easy to use and can answer questions quickly and accurately. These results underscore the potential of the RAG system in transforming the educational process by offering an efficient and interactive learning and literature comprehension experience. This research highlights the need for further refinements in such systems, emphasizing their importance in educational settings.

Keywords: *Retrieval-Augmented Generation (RAG), Retrieval-based, Generative models, Student Learning*

1. INTRODUCTION

The development of the use of digital technology, the internet, and computers in Indonesia today has drastically changed and transformed how students find knowledge at university. One of the transformational impacts of technological development is the ease of access to information. Along with the development of the internet and technology, students can easily access millions of electronic-based resources such as electronic books, electronic scientific journals, electronic theses, articles, online lecture materials, and others. Students are no longer limited to campus library collections or physical printed books, which are limited in number [1], [2]. Technological advancements have also brought innovations in student learning on campus. Students can now access online learning platforms such as Learning Management Systems (LMS) and online materials that can be downloaded and used for independent learning. They can freely utilize digital information resources, such as e-books, electronic journals, and online learning materials available on LMS services [3]. However, the ease of finding and accessing knowledge sources does not necessarily increase reading interest or literacy. Indonesian students' interest and ability to learn and read has been assessed through various metrics, with the Program for International Student Assessment (PISA) finding striking results where Indonesia ranked 7th out of ten countries with the lowest overall PISA score in 2023. It is, therefore, a cause for concern about literacy levels in Indonesia. For example, a decline in literacy levels among Indonesian students was noted, which correlates with a decline in PISA rankings over time. Specifically, from PISA 2015, Indonesian students' reading competency scores dropped from 397 to 371. In addition, it was highlighted that Indonesia typically ranks low, often among the bottom ten, despite a potential upward trend in the 2022 or 2023 index [4], [5].

BINUS University is one of the leading private universities in Indonesia. BINUS University itself has successfully created a more integrated and technology-based learning environment. Students can now access and learn learning materials online through a Learning Management System (LMS)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3]. LMS is a digital platform specifically designed to support the learning process. Through LMS, students can access various learning materials online. This platform provides e-pdf and e-documents containing lecture materials, modules, reading materials, and other reference sources. With these materials in digital format, students can easily download and access them anytime and anywhere, even outside lecture hours. This allows students to access quality digital learning resources easily and provides great flexibility in exploring and learning materials independently. BINUS University also has a web-based online library service. This service is specifically designed to fulfill students' information needs by providing reading materials in digital form, such as e-books, undergraduate e-theses, graduate e-theses, e-journals, e-research, and many more. This online library service allows students to view, read, borrow, and download various digital materials according to their needs [3], [6]. Through this online library, BINUS University aims to provide accessibility and convenience for students to discover knowledge. With easy access to digital materials, students at Bina Nusantara University can develop a deep understanding of their field of study and maintain the novelty of knowledge. We can see that there is a transformation, where books or learning documents that were previously in physical form become digital. Learning literature that is usually located in universities, libraries, and bookstores has also undergone a transformation, which is now on the Internet so that it is easy to access. However, there is one thing that has not changed until now, namely the way students learn from literature. Where to understand the contents of the literature, students must read the entire contents of the literature. This causes several problems, such as language barriers where students will find it difficult to understand the contents of foreign-language literature and it also takes longer to understand the contents of the literature. The way of learning literature by reading is also very boring and not interactive.

Recently, a new technology concept called Retrieval-Augmented Generation (RAG) has emerged. RAG is a technique in natural language processing (NLP) that combines the strengths of retrieval-based models and generative models, such as the Large Language Model (LLM), to improve the quality and relevance of the generated text. Retrieval-based models are good at finding relevant information from a large corpus of text. In contrast, generative models are good at generating new text that is consistent with the given input [7]. By combining these two approaches, RAG can produce more informative and coherent text than either approach alone. RAG is well suited for tasks that require factual accuracy and creativity, such as question answering, summarizing, and story writing. In question answering, for example, RAG can first use a search-based model to find relevant passages or documents containing answers and then use an LLM-capable generative model to generate concise and coherent responses based on that information and multiple languages.

This is an excellent opportunity where RAG can change the way students understand and learn from literature. Students will be able to directly ask questions interactively, summarize, and understand the content of the literature [7]. This can help overcome the previously mentioned problems. Therefore, the contribution of this paper is that we propose to build a Retrieval-Augmented Generation (RAG) System for enhanced student learning. After this prototype is built, testing and evaluation will be carried out using black box and usability testing, which several BINUS University students will carry out.

2. STATE OF THE ART

2.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a concept in the field of artificial intelligence and natural language processing that combines two main components: generative models (such as Large Language Models, or LLMs) and retrieval systems (search systems, usually vector databases) [7]. The way RAG works can be explained through several main steps:

- Question Processing, where when a question or request is given to the RAG system, the first step is to process the question using LLM. LLM can understand and analyze the question based on the context and nuances of the language.
- Retrieval, after understanding the question, the system then uses its trained retrieval model to search for answers or related information from a database. This database usually consists of documents or data that have been indexed in vector form. This search usually uses techniques such as similarity search, where the vectors of the queries are compared with the vectors of the documents in the database to find the best match.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Information Integration, after finding relevant information, the system then combines that information with LLM's internal knowledge. This allows the model to generate more accurate and informative answers, as it relies not only on its own knowledge, but also relevant external information.
- Answer Generation, Finally, by combining information from the retrieval system and LLM internal knowledge, the RAG system generates a comprehensive answer or output. This output is not just based on the retrieved data, but also how the LLM understands, interprets, and integrates that data in the context of the question.

The use of Retrieval-Augmented Generation (RAG) technology opens many possibilities for various applications that require advanced natural language understanding and processing. Some potential applications of RAG include:

- Advanced Search and QA Systems: RAG can be used to develop more advanced search and question-answering (QA) systems. These systems can provide more accurate and detailed answers to complex questions, by combining knowledge from large databases and language understanding capabilities from generative models [7].
- Virtual Assistants and Chatbots: In the development of virtual assistants and chatbots, RAG enables the creation of more natural, informative, responsive, and interactive dialogs. This is useful for customer service, education, and entertainment applications [7].

2.2 Large Language Model (LLM)

LLM is a type of machine learning algorithm designed to understand and generate text in human language. LLM works by analyzing a large amount of existing text in a human language and learning the structure, grammar, and patterns contained in the text. Once trained, LLMs can be used for a variety of tasks, including translation, text generation, question answering, and more. LLM uses complex deep learning neural network architectures, such as Transformer, to understand and generate text. The Transformer architecture is known for its ability to overcome issues related to long distances in text and produce robust text representations [8]. There are several LLM models in development today, such as GPT-4 (Generative Pre-trained Transformer 4) which is the latest iteration of the GPT family of models. This model has greater capacity and can produce higher quality text compared to previous versions. GPT-4 has been trained on larger data and can perform various language tasks with a high level of artificial intelligence [9]. There is the BERT model (Bidirectional Encoder Representations from Transformers) BERT is another very popular LLM model. One of the main advantages of BERT is its ability to understand context better through understanding words in their actual context. It has been used extensively in tasks such as natural language understanding, information mining, and sentiment analysis. There is also the XLNet model, this model overcomes some of the limitations of GPT by introducing a permutation-based approach to training. This allows XLNet to understand more complex inter-word dependencies [10]. And there are many more rapidly developing models such as Llama 2 developed by meta and Gemini developed by Google, and many more.

2.3 LangChain

LangChain is a framework for developing applications powered by large language models (LLM). LangChain makes it possible to build context-oriented applications by connecting language models with other sources of context (prompt instructions, multiple examples, content to base the response on, etc.) [11]. LangChain is a framework designed to enable question-answering applications over various types of documents like PDFs, blogs, and Notion pages. It leverages Large Language Models (LLMs) for their ability to understand and process text. Here's an overview of how LangChain facilitates this:

- Loading: Data, such as documents, are loaded into the system.
- Splitting: These documents are then broken down into smaller parts or splits.
- Storage: The splits are stored, often in a vector store, which may also embed the splits.
- Retrieval: The system retrieves splits from storage that are relevant to the input question, usually based on similar embeddings.
- Generation: An LLM generates an answer using a prompt that includes both the question and the retrieved data.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LangChain facilitates the creation of Retrieval-Augmented Generation (RAG) systems by streamlining the process of integrating different components like document loaders, splitters, storage systems, and language models into a cohesive question-answering pipeline. This integration makes it simpler to build powerful and efficient RAG systems that can leverage the vast information available in various document types to provide detailed and contextually relevant answers.

2.3.1 ChatGPT

ChatGPT, an innovative language model, empowers users to interact with computers in a more conversational and natural way. It takes its name from "Generative Pre-trained Transformer," which is a class of natural language models developed by the open-source Artificial Intelligence (AI) community. The hallmark of generative AI, the term that defines this form of AI, is its ability to generate native output [9]. ChatGPT uses natural language processing to assimilate and learn from huge volumes of internet data, thus generating AI-based textual responses, answers, and solutions. These models undergo rigorous training on vast text datasets to anticipate the next words in a sentence, leading to the creation of coherent and convincing human-like responses to questions or statements. ChatGPT, for example, relies on 570 GB of data, consisting of 300 billion words, and includes approximately 175 billion parameters. ChatGPT's easy-to-use interface allows it to be thought of as a computer robot capable of understanding and discussing any topic. It can provide data, analysis, or even opinions when requested. Nonetheless, its algorithms have no definitive point of view, as its interpretations are entirely statistical, based on analyzing billions of texts found on the internet. The version of ChatGPT is built on GPT-3.5, which is the latest free version accessible today, while a more advanced version, capable of interpreting different types of data and equipped with better writing capabilities, is expected to appear in the future.

2.4 Semantic Search

Semantic search is an advanced search methodology that utilizes natural language processing to understand user intent and the context of search queries. Unlike conventional search engines that rely on keyword matching to provide results, semantic search engines use algorithms to understand the meaning of the query and the search context to provide relevant search results [12]. The basic mechanism of Semantic search is to scrutinize the user's query and assess the search context. It carefully examines the words used in the query, the relationship between them, and the user's intent. For example, when a user searches for "best restaurants in New York City," the semantic search engine understands that the user is interested in restaurants in New York City, not other business categories. Based on this understanding, the engine will return results that are more relevant to the user's intent. These semantic search engines can examine the relationship between words in a query to figure out the user's intent. For example, if a user searches for "best Italian restaurants in New York City," a semantic search engine understands that the user is interested in Italian restaurants rather than all types of restaurants. The search engine then provides search results that better match the user's intent. Semantic search engines assess the user's intent while scrutinizing the query. For example, when a user searches for "best Italian restaurants in New York City," the semantic search engine identifies that the user is interested in Italian restaurants rather than all types of restaurants. It then returns search results that are more relevant to the user's intent. Semantic search engines evaluate the context of the search to provide more targeted results. For example, when a user searches for "best Italian restaurants in New York City," the semantic search engine understands that the user is primarily interested in Italian restaurants in New York City rather than other types of restaurants in any city. Therefore, this search engine returns search results that are more relevant to the user's intent.

2.5 Embedding

The concept of embedding refers to a collection of vector representations of text and code that have been developed by OpenAI. These embeddings, or their features, can be utilized in various applications such as text similarity computation, semantic search, and text classification. OpenAI Embedding is a cutting-edge machine learning approach that involves pre-training models on unsupervised data using a technique called contrastive pre-training. This methodological approach ensures that the model creates a vector representation of text or code that can recognize patterns independently and can then be used as features in various applications [9], [13]. The vector representation of text and code in OpenAI Embeddings is generated using unsupervised training, a process that trains the model to recognize data patterns without explicitly giving it any categories or labels to learn from. Afterwards, the model is tested using supervised

Journal of Theoretical and Applied Information Technology
30th November 2023. Vol.101. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

data to assess its performance. When appropriate performance is achieved, the vector representation can be used as a feature in other applications. OpenAI Embeddings is a versatile tool that can be used in various applications, including but not limited to, semantic search, text classification, and text similarity calculation. Utilizing these features in various applications will increase accuracy, accelerating the realization of accurate and timely projects. Vector representations can also be used for data visualization, making data analysis easier and more efficient.

2.6 Vector Database

Vector databases use a different approach than traditional databases to process and optimize data. While conventional databases store scaled data types such as numbers and strings in rows and columns, vector databases operate on vectors. As a result, querying and optimization differ significantly from traditional databases. To search rows in a traditional database, we usually query for values that match our search criteria. On the other hand, it uses similarity metrics to find the vector that is most like our query. Vector databases apply a combination of various algorithms that participate in Approximate Nearest Neighbor (ANN) search [14]. These algorithms improve search optimization through procedures such as hashing, quantization, or graph-based search, which are assimilated into a pipeline. This pipeline ensures fast and accurate retrieval of the neighbors of the queried vector. In vector databases, the main trade-off is between speed and accuracy, where greater accuracy results in slower queries. However, sophisticated systems can provide very fast search results with almost perfect accuracy.

2.7 Related work

Paper entitled "Retrieval-Augmented Generation Question Answering for Event Argument Extraction" proposes a breakthrough framework known as R-GQA, which combines a retrieval-augmented mechanism with generative question answering for event argument extraction from text. This method seeks to overcome the shortcomings of both extractive approaches and purely generative approaches, which are commonly used in traditional event argument extraction. The R-GQA framework operates by taking relevant question-answer pairs and using them as additional context to guide the argument extraction process. The approach leverages pre-trained language models and a novel clustering-based sampling strategy, JointEnc, to improve learning with less retrieval. The empirical studies in this paper, which include fully supervised learning, domain transfer, and learning with few shots' scenarios, demonstrate the superiority of the R-GQA model compared to traditional methods. The results highlight significant progress in terms of performance and efficiency, especially in complex scenarios such as domain transfer and learning with few shots, thus marking an important contribution in the fields of natural language processing and event argument extraction [15].

Paper entitled "Lift Yourself Up: Retrieval-augmented Text Generation with Self-Memory," the authors introduce Selfmem, an innovative retrieval-augmented text generation framework designed to enhance the capabilities of generation models. Unlike traditional models that rely on fixed external memory sources, Selfmem employs its own outputs as a dynamic, evolving memory pool. This approach, termed self-memory, involves a retrieval-augmented generator for sourcing memory from a datastore and a memory selector for choosing the most suitable outputs for future generation rounds. The framework demonstrates remarkable improvements in text generation tasks such as neural machine translation, abstractive text summarization, and dialogue generation. By using its own generated content as a reference, Selfmem not only achieves state-of-the-art results but also addresses limitations of fixed-memory retrieval, marking a significant advancement in the field of natural language processing and text generation [7].

In this research, the system uniquely integrates GPT and LangChain, setting it apart from traditional models. This integration facilitates a more nuanced understanding and generation of language, enhancing the system's ability to provide contextually relevant and accurate responses. Also, using customized prompting engineering to GPT to specifically address the challenge of language barriers and passive reading methods in educational settings. This focus is particularly relevant in the diverse linguistic landscape of BINUS University, where students often face difficulties in engaging with content in a second language. Also the system fosters interactive learning through dynamic question-answering sessions, a feature less emphasized in other systems. This interactivity is crucial in engaging students more deeply with the material and active learning.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. METHODOLOGY

3.1 Research Methodology

In this research, several stages are carried out, namely by collecting data first by conducting observations and literature studies as part of the State of the art through journals, books, and information on the internet. Then, designing the concept of the Retrieval-Augmented Generation system prototype. Followed by designing concept design and architectural design. After that, continued with Implementation by building a prototype of the Retrieval-Augmented Generation system and then evaluating it by conducting Interview and Black box and usability testing conducted on several BINUS students to find out whether the system works well and accordingly.

3.2 Concept Design

Retrieval-Augmented Generation (RAG) system in this research will be web-based application on localhost. design concept of the system has several features such as Sign-in/Sign-up, Literature page view, and Retrieval-Augmented Generation:

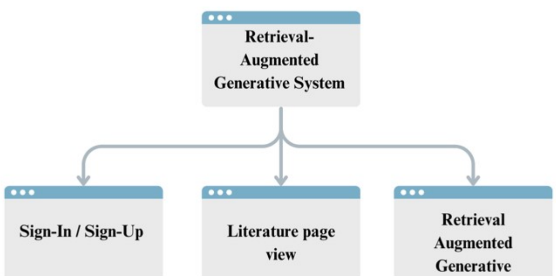

Figure 1: Concept Design Retrieval-Augmented Generation (RAG) system

- Retrieval-Augmented Generation (RAG), is the main feature of this system where this feature can be used by students to conduct interactive learning such as chat[7]. Students can ask about the context in the e-pdf digital document and RAG will provide answers according to the context in the digital document or literature. This feature specifically helps students in learning and understanding literature reading more effectively and enjoyably Students, and can also determine how the answers given by the RAG system by entering query prompts such as: What is the summary of this file, explain, and answer in Spanish, etc. RAG can provide answers that are fast, interactive, and capable of understanding multiple languages.
- Sign-in/Sign-up, is the feature of logging into the Retrieval-Augmented Generation system application. Students who have signed-in can do literature learning in it.
- Literature Page view is a feature where in the application there will be a section to view the contents of the literature that has been entered, so students can still do literature learning as usual by reading the contents of the literature.

3.3 Architecture Design

In this section, we will explain the architecture of each feature in the Retrieval-Augmented Generation (RAG) system and how the architecture works and produces an output. The technology used is also explained.

3.3.1 Retrieval-Augmented Generation (RAG) Architecture

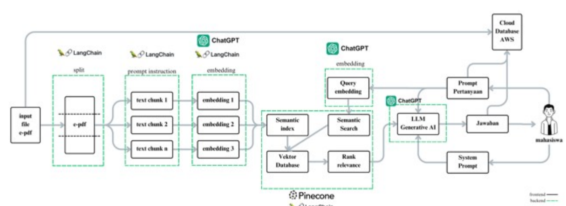

Figure 2: Retrieval-Augmented Generation (RAG) Architecture

Figure 2 shows displays the architecture of the RAG feature, starting with students inputting digital document files in the form of e-pdf and filling in questions and filling out system prompts. Next, using tools in the LangChain framework such as document Loaders to extract the contents of the e-pdf file and split the document using the text splitter tool. Split documents are done to produce text chunks and each chunk size contains 1000 words. Next use embedding on ChatGPT to embed each text chunk. Embedding is done to convert text into text as vector and build semantic index. Then send the text as vector to the cloud database of pinecone called vector database. Furthermore, student questions will be converted into text as vector using chatgpt embedding. Text as vector from questions will be semantic search or search on text as vector in vector database based on relevance

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and similarity of meaning between text as vector. Furthermore, the text as vector will be converted back into a series of text using chatgpt and langchain and then processed using LLM Genarative AI such as chatgpt to produce a language that can be understood by humans [16]. Furthermore, the System prompt is used to dictate how the answer results and then the output will be in the form of text answers. the answer results and question prompt and the inputted e-pdf files will be stored using the AWS cloud database.

3.3.2 Sign-in/Sign-up Architecture

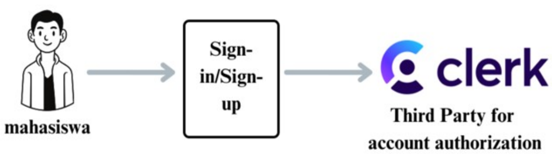

Figure 3: Sign-in/Sign-up Architecture

Figure 3 shows the Sign-in/Sign-up architecture, students can either Sign-in or Sign-up where we use a third party for Clerk account authorization. Clerk is a third-party service that provides more than user authentication and provides everything needed to manage user onboarding and allow them to manage their accounts. This includes an optimized and fully customizable login experience. Clerk allows the selection of authentication strategies, including passwords, email codes or links, OAuth, and more.

3.3.3 Literature Page view Architecture

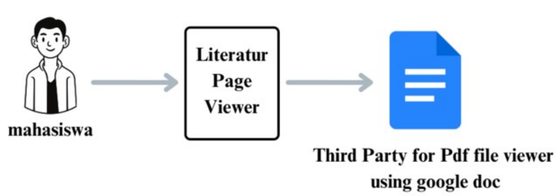

Figure 4: Literature Page view Architecture

Figure 3 shows the display architecture of the literature page, this section will display the contents of the literature. With this, students can use it to learn the contents of the literature by reading the literature.

3.4 Implementation

3.4.1 Prerequisites

Before starting to develop the application, it is necessary to check whether all the prerequisites are installed on the platform where the application system will be developed. The system should have installed dependency tools such as npm, node, python3, langchain, and nextjs. And for the development environment, the application system will use VS Code and Google Chrome.

3.4.2 ChatGPT

The development of the application system also requires the use of the Chatgpt model 4 turbo API. To get the API, you can sign-up on platform.openai.com and in the personal section there is a view api key, enter the view api key and generate api key to get the API key from chatgpt model 4 turbo. However, for the Retrieval-Augmented Generation system to run properly, a paid version of the chatgpt model 4 turbe API is required by depositing a minimum of 5 dollars. After making a deposit with a set payment, then we can use the API smoothly on the application system.

3.4.3 Pinecone

In the development of the application system, it is also necessary to use the API, index name, and Environment from Pinecone's vector database. To get the API, Index name, and environment on Pinecone, you need to sign up pinecone.io. Next create index and fill in the desired index name, select metric cosine and enter dimensions 1536. Then we will get the API, index name, and Environment and ready to use [14].

3.4.4 Cloud Database AWS

In the development of the application system, it is also necessary to use the API from the AWS cloud database which will be used to store literature files and store chat history from RAG, index name, and Environment from Pinecone's vector database. To get the API, and the environment on AWS, you need to sign up at aws.amazon.com, then go to the s3 (simple storage service) menu and start with the create bucket, set the AWS region to determine the server environment and the API will be created.

3.4.5 Web Application

Application development will be in the form of a web on localhost. using Python3, LangChain, javascript, reactjs, nextjs and some use of APIs from third parties for this web application. The database used is an external database in the form of a vector database from Pinecone and a database from AWS.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.5 Evaluation

After the Retrieval-Augmented Generation (RAG) system has been developed, it will be evaluated by conducting two types of testing, namely Blackbox Testing and Usability Testing. This testing will be carried out by several BINUS students, and it is hoped that through the implementation of these two types of testing, holistic evaluation results will be obtained. Blackbox Testing aims to identify potential technical and functional problems in the system being evaluated, while Usability Testing will provide an in-depth understanding of the user experience and the extent to which this product can be used with ease and efficiency by end users. The combination of these two testing approaches is expected to provide a more comprehensive picture of the quality and performance of the product or system, as well as being able to assist in identifying potential improvements needed to enhance the user experience.

4. RESULTS AND DISCUSSION

4.1 Implementation Result

The results of the implementation in the form of Retrieval-Augmented Generation system have been successfully made and can be used locally. The following is a view of the Retrieval-Augmented Generation system, where there is a landing page, the user can press the "Start | Your Leaning Space" button, and the user will enter the sign in / sign up page using a google account. Furthermore, when the user has logged in, there will be a home page display, where the user inputs the literature file for the learner to do. After the user inputs the literature file, the user will automatically move the page to the learning space page. On the learning space page, the user can read the literature because there is a display of the content of the literature and ask questions about the context in the literature by using the interactive chatbox "Chat to your Literature" which is the main feature of RAG. In addition to users being able to explore the context in the literature, and finally on the learning space page, users can save literature reading material to make it easier for users to learn literature.

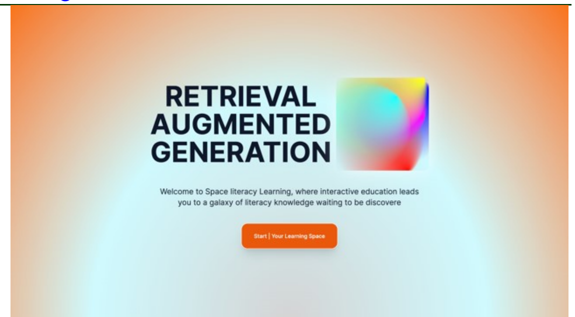

Figure 5: Landing Page

Landing page is the initial display when students enter the RAG application, users can press the main button and users will be redirected to the Sign-in / Sign-up page.

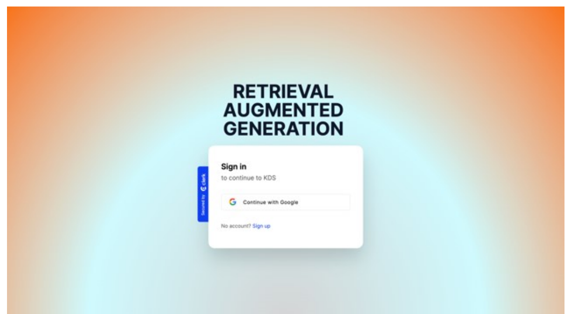

Figure 6: Sign-in/Sign-up Page

On the Sign-in/Sign-up page, students can register an account only by using their Google account, after which they will be redirected to the next page, namely the home page.

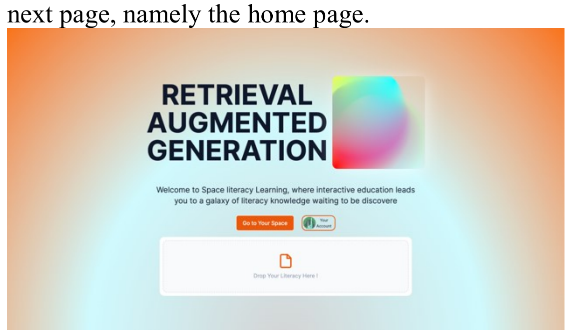

Figure 7: Home Page

On the home page students can use several such as Your Account which functions to check account details and logout. There is also a document icon where students click and can input digital literature document files in pdf format that they want to study and will immediately switch to the Learning Space Page. And finally, there is the Go to Your Space button which functions to redirect students to the next page, namely the Learning Space Page.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

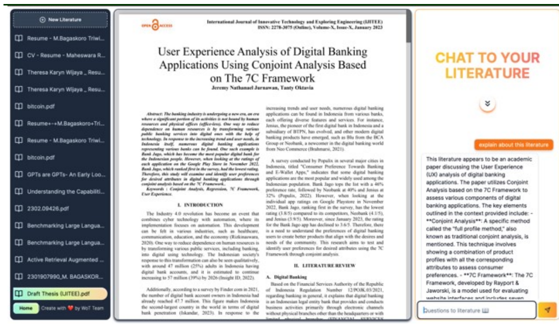

Figure 8: Learning Space Page

This learning space page will be the main page for students to study and learn from their literature documents. Where there are 3 sections. The leftmost part is the history of literature document files that have been inputted, where files stored there will be able to be displayed again without the need to re-input the file. In the middle there is a literature page view, students can freely do literature learning by seeing and reading the entire contents of the literature document. And the rightmost is the main feature of RAG called Chat to your literature, this feature can be used for students to ask about the context in the literature document, summarize the contents of the literature, and of course it can all be done in an interactive way like a question-and-answer chat.

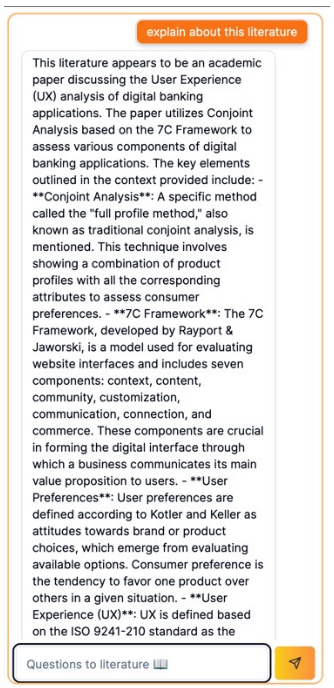

Figure 9: RAG chat feature

Figure 9 shows the results produced by the RAG feature, in the image there is a question to explain about the literature document entered and RAG will quickly answer the question. This RAG feature will be able to be used by students to learn and understand literature more interactively.

4.2 Black box Result

To see whether the RAG prototype application system is running properly, Blackbox testing was carried out, with the following results:

Blackbox testing on the landing page the results obtained were successful in all scenarios.

Table 1: Blackbox testing on Landing Page

Landing Page				
No	Scenario Expected	result	Test Result	Status
1	Landing Page	The appearance of the landing page is complete and there is a main button (Start | Your Learning Space)	Success	Valid
2	Button (Start | Your Learning Space)	Button can be pressed and successfully bring to the sign-In / sign-up page	Success	Valid

Blackbox testing on the Sign-in/Sign-up Page the results obtained were successful in all scenarios.

Table 2: Blackbox testing on Sign-in/Sign-up Page

Sign-in/Sign-up Page				
No	Scenario Expected	result	Test Result	Status
1	Sign-in and Sign-up page	Sign-in and Sign-up page appears	Success	Valid
2	Sign-in via Google	The login page appears, select a google account	Success	Valid
3	Sign-up via Google	The registration page for your account appears	Success	Valid

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Blackbox testing on the Home Page the results obtained were successful in all scenarios.

Table 3: Blackbox testing on Home Page

Home Page				
No	Scenario Expected	result	Test Result	Status
1	Home Page	the home page complete with account and drop box appears	Success	Valid
2	Icon Account	The account icon that is pressed will appear google account info	Success	Valid
3	Drop box (Drop Your Literacy Here)	Drop box can be pressed will bring up the file location, the user selects the pdf file that will be input. Next will bring up the Learning Space page	Success	Valid

Blackbox testing on the Learning Space Page the results obtained were successful in all scenarios.

Table 4: Blackbox testing on Learning Space Page

Learning Space Page				
No	Scenario Expected	result	Test Result	Status
1	Learning Space page	learning space Page appear	Success	Valid
2	Sidebar Document	The appearance of the document's sidebar, the user can select the document file that has been inputted and will appear dokumen pdf view	Success	Valid
3	PDF document view	The pdf view document that the user will use to read appears	Success	Valid
4	RAG chatbox	User gets answer from RAG from user's question	Success	Valid
5	Home Button	Home button can be pressed and will bring up the home page	Success	Valid
6	New Literacy Button	Can be pressed and will bring up the home page	Success	Valid

4.3 Usability Result

To measure the feasibility level of the appearance (UI), as well as the function of the RAG feature called "Chat to Your Literature", and the ease of use of the RAG application system, usability testing was carried out by involving students from BINUS University directly and the results:

Usability testing conducted to BINUS students to test the proficiency of the RAG User Interface (UI) showed good results in all aspects of the test.

Table 5: Usability testing on User Interface (UI)

No	Test Aspect	Description	Result
1	Visual consistency	The UI display has good visual consistency, with the use of uniform color and design patterns throughout the application.	Good
2	Clear display of information	clear and well-structured information display. Data and content are neatly displayed	Good
3	Intuitive navigation	The navigation within the KDS app is intuitive and easy to understand and can quickly find the function they are looking for without any difficulty.	Good
4	Readability of text	The text and fonts used are easy to read and there is no difficulty in	Good

		identifying the information.	
5	Clear use of icons	The icons used in the app are considered very representative and help users to quickly understand certain functions.	Good
6	Spacing	The UI provides enough space between elements, is not too dense, thus avoiding confusion and allowing users to interact comfortably.	Good

Usability testing conducted on BINUS students to test the function of the RAG chat feature showed good results in 5 test aspects and 1 test aspect that showed not good results.

Table 6: Usability testing on Functionality RAG

No	Test Aspect	Description	Result
1	Interactive Answers	RAG chat feature can do interactive answers quickly	Good
2	Inside Context Answer	The answers generated by RAG match those in the pdf context	Good
3	Answer feedback "In Context"	In the answers provided by the RAG chat feature, the information provided is accurate according to the context of the literature document.	Good
4	Inaccurate answers when answering follow-up questions	Sometimes, when RAG chat is asked the next question, the answer given is still related to the previous question before finally answering the new question. However, if asked another question, it may not respond well and repeat the previous answer.	Not Good
6	Use of Different Language	The answers generated by RAG chat can use multiple languages such as English, etc. and languages. So that it helps students to learn from literature or reading sources from many languages	Good

4.4 Discussion

From the research conducted, it shows that a prototype implementation of the Generative Retrieval-Augmented system was successfully developed and can also be used locally. The system also has an easy-to-use interface, starting with a home page where users can begin their journey by pressing the "Start | Your Learning Space" button. This design choice shows the focus on ease of use and accessibility. While the results obtained from conducting blackbox testing show that this application can work well and has no problems on each page. Usability testing to test the User Interface also showed good results from all aspects of the UI tested. In usability testing to test its functionality, 5 out of 6 aspects tested showed good results but 1 aspect was found to be less good where sometimes when RAG chat is given the next question, the answer given is still related to the previous question before finally answering the new question. However, if given another question, it may not respond well and repeat the previous answer.

Also, in this research, this RAG system is distinguished from existing literature by focusing on interactive learning and overcoming language barriers, a significant deviation from studies primarily centered on information retrieval. Our methodological innovation lies in the integration of ChatGPT model and LangChain, facilitating an interactive and multilingual learning environment unique to our setting at BINUS University. And results of our research demonstrate a notable improvement in student engagement and comprehension, showcasing the effectiveness of our system in an educational context. This research not only contributes innovatively to the field of educational technology but also opens avenues for future studies to explore the application of similar systems in diverse educational settings and disciplines, underscoring the adaptability and broad applicability of our approach.

4.5 Limitations

The testing that has been conducted shows that only a few students have been tested.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Therefore, in the future, more students can be tested to validate the effectiveness of the RAG system and to ensure that the user interface design has an impact in increasing students' interest in learning. Moreover, although the integration of technologies such as ChatGPT marks a significant step forward, there are some inherent limitations to this technology, namely the need to improve the prompting engineer in the GPT model. Prompting engineers is very important for the functionality of the RAG system to work properly. In the results of usability testing on functionality RAG can be seen in number 4 the results are still not good. This is because the engineer prompt in the GPT model is not perfect. Therefore, in the future, this system will continue to test and improve the engineer prompt to produce even better functionality.

5. CONCLUSION

This research successfully demonstrated the success in building a Retrieval-Augmented Generation (RAG) system. Which can enhance the learning experience of Private University students in studying and understanding literature. The RAG system, a web-based application developed in this research, offers a revolutionary approach to learning, allowing students to interactively engage with literature and educational materials. This research addresses the challenges students face in accessing and understanding digital literature due to language barriers and passive reading. Through this RAG system, students can learn and understand literature quickly, interestingly, and more interactively.

System evaluation through black box testing and usability testing involving several BINUS University students gave positive results. The system demonstrated strong functionality across a range of features, with the RAG chat feature being particularly effective in providing interactive and contextually appropriate answers. Although the system demonstrated high levels of usability and user interface satisfaction, there were some limitations identified, particularly in the handling of the RAG chat feature for follow-up questions. These insights are valuable for further refinement of the system.

For future research and development in this area. As technology develops, systems such as this are expected to play an increasingly important role in educational environments, offering a personalized, efficient, and more interactive learning experience.

REFERENCES:

[1] A. Annisa, “Sejarah Revolusi Industri dari 1.0 sampai 4.0 Artikel Mahasiswa Sistem Telekomunikasi View project”, doi: 10.13140/RG.2.2.20215.24488.

[2] M. L. Gueye and E. Exposito, “University 4.0: The Industry 4.0 paradigm applied to Education.” [Online]. Available: https://hal-univ-pau.archives-ouvertes.fr/hal-02957371

[3] K. Iskandar, D. Thedy, J. Alfred, and Yonathan, “Evaluating a Learning Management System for BINUS International School Serpong,” in *Procedia Computer Science*, Elsevier, 2015, pp. 205–213. doi: 10.1016/j.procs.2015.07.556.

[4] datapandas.org, “PISA Scores By Country.”

[5] R. Febrian and Y. Mahabarata, “Literacy Emergency Among Indonesian Students.”

[6] M. Wiannastiti, “HOW TO TEACH ENGLISH ENTRANT FOR BINUS UNIVERSITY STUDENTS USING A CELL GROUP METHOD SUPPORTED BY BINUSMAYA,” 2011.

[7] X. Cheng, D. Luo, X. Chen, L. Liu, D. Zhao, and R. Yan, “Lift Yourself Up: Retrieval-augmented Text Generation with Self-Memory.”

[8] O. Topsakal and T. C. Akinci, “Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast,” *International Conference on Applied Engineering and Natural Sciences*, vol. 1, no. 1, 2023, doi: 10.59287/icaens.1127.

[9] OpenAI, “GPT-4 Technical Report,” Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.08774

[10] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized autoregressive pretraining for language understanding,” in *Advances in Neural Information Processing Systems*, 2019.

[11] O. Topsakal and T. C. Akinci, “Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast,” *International Conference on Applied Engineering and Natural Sciences*, vol. 1, no. 1, 2023, doi: 10.59287/icaens.1127.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[12] S. Radha, A. Ramachandran, and R. Sujatha, "Semantic search engine: A survey." [Online]. Available: www.ijcta.com

[13] A. Neelakantan *et al.*, "Text and Code Embeddings by Contrastive Pre-Training," Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.10005

[14] "AI assistant for document management Using Lang Chain and Pinecone," *International Research Journal of Modernization in Engineering Technology and Science*, 2023, doi: 10.56726/irjmets42630.

[15] X. Du and H. Ji, "Retrieval-Augmented Generative Question Answering for Event Argument Extraction," Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.07067

[16] "LangChain-Powered Virtual Assistant for PDF Communication," *International Research Journal of Modernization in Engineering Technology and Science*, 2023, doi: 10.56726/irjmets43587.