Chatbots in Academia: A Retrieval-Augmented Generation Approach for Improved Efficient Information Access

1st Maryamah Maryamah
Data Science Technology
Faculty of Advanced Technology and Multidiscipline
Universitas Airlangga
Surabaya, Indonesia
maryamah@ftmm.unair.ac.id

2nd Muhammad Maula Irfani
Data Science Technology
Faculty of Advanced Technology and Multidiscipline
Universitas Airlangga
Surabaya, Indonesia
muhammad.maula.fani-2021@ftmm.unair.ac.id

3rd Edric Boby Tri Raharjo
Data Science Technology
Faculty of Advanced Technology and Multidiscipline
Universitas Airlangga
Surabaya, Indonesia
edric.boby.tri-2022@ftmm.unair.ac.id

4th Netri Alia Rahmi
Data Science Technology
Faculty of Advanced Technology and Multidiscipline
Universitas Airlangga
Surabaya, Indonesia
netri.alia.rahmi-2021@ftmm.unair.ac.id

5th Mohammad Ghani
Data Science Technology
Faculty of Advanced Technology and Multidiscipline
Universitas Airlangga
Surabaya, Indonesia
mohammad.ghani@ftmm.unair.ac.id

6th Indra Kharisma Raharjana
Informations Systems
Faculty of Science and Technology
Universitas Airlangga
Surabaya, Indonesia
indra-k-r@fsaintek.unair.ac.id

2024 16th International Conference on Knowledge and Smart Technology (KST) | 979-8-3503-7073-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/KST61284.2024.10499652

***Abstract*—In today's digital age, higher education utilizes chatbots as virtual assistants to assist users, especially prospective students to access information easily. A chatbot is an application in natural language conversations to simulate intelligent interactions. Intelligent chatbots are needed to understand user needs and answer questions relevantly. We propose a chatbot with Retrieval Augmented Generation approach involving a retriever with cosine similarity search using OpenAI Ada embeddings to obtain relevant documents. The LLM OpenAI GPT-3.5-Turbo then generates the final answer. The chatbot mechanism begins with the retrieval module systematically identifying documents stored in the vector database that contain relevant information related to the user's query. The selected documents and query are provided to the LLM as part of the prompt to generate responses based on the knowledge provided in the relevant documents. The retrieval method is evaluated based on two criteria: the search method and the embedding model. The comparison method uses similarity search with Maximum Marginal Relevance (MMR) Search and the proposed embedding method against other models such as Google Embedding-001 and MPNet-Multilingual. The retrieval process is assessed using an evaluation dataset that incorporates Recall and Precision metrics, while answer generation is measured with BLEU and ROUGE Score. The observed disparity result between similarity search and MMR is not notably significant. Nonetheless, our chatbot holds an advantage in referencing past conversations due to its ability to store conversation history. Furthermore, potential enhancements are identified by augmenting the knowledge provided to the LLM in forthcoming iterations.**

Keywords*— Academic Chatbots, Retrieval-Augmented Generation*, *Large Language Models*, *Technology

I. INTRODUCTION

In the current age, the strive for information access is undeniable. Higher education institutions are now utilizing chatbots to provide support to users. This facility is particularly valuable for prospective students who may enroll. A chatbot is an application that engages in natural language conversations to simulate intelligent interactions with users [1], [2], [3]. Chatbots employ Natural Language Processing (NLP) techniques to provide users with a convenient means of obtaining information through interactive and conversational inquiries [2].

The development of chatbots before the release of OpenAI GPT-3.5 commonly used intent-based classification methods [4], [5]. In this approach, a dataset containing pairs of questions and answers, along with tags or intents related to the topics of those answers, was typically used. In intent-based chatbots, when a user inputs a question into the chatbot, it goes through a classification model to predict the most relevant tag or intent for that question. Afterward, the chatbot responds with a predefined answer that corresponds to the tag found in the dataset. The classification-based method has a drawback when user inputs a question that does not match any of the tags in the dataset, the chatbot will provide an irrelevant and uninformative response. Additionally, if a user sends a follow-up question, the chatbot cannot refer to previous conversations because it lacks memory.

In addition to intent-based classification methods, there are generative-based methods [6], [7]. In this approach, chatbots can provide responses that are not discrete, allowing for more natural conversations. However, before the release of powerful Large Language Models like OpenAI GPT-3.5 and frameworks like LangChain that simplify the development of applications based on LLMs, generative methods typically relied on deep learning models for sequence data, such as LSTM (Long Short-Term Memory) networks. However, creating a generative model from scratch requires significant resources and a large dataset.

The development of our chatbot employs the Retrieval-Augmented Generation (RAG) technique which combines pre-trained parametric and non-parametric memory for response generation [8]. RAG consists of a Retrieval module and an LLM-based generation module [9]. Using a pretrained LLM, the model can perform as well as an LLM trained on a large-scale dataset, which generally performs well. In addition, when using Large Language Models (LLMs) available through APIs like those provided by OpenAI and Google Gemini, the computing resources for the trained models are not needed. However, choosing closed-source LLMs has drawbacks in terms of costs. Extra knowledge from university documents, websites, or related sources needs to be included and prompted to the LLM. However,

979-8-3503-7073-7/24/$31.00 ©2024 IEEE

since LLMs typically have a limited context size, the most relevant documents must be selected before being prompted as context. RAG's retriever is responsible for selecting relevant documents and the LLM can answer questions in an appropriate context with efficient token usage. Overall, the purpose of RAG is to improve the question-answering capabilities of the chatbot. In this study, the combination of OpenAI GPT-3.5 is used as the Large Language Model (LLM) for generating answers and OpenAI Ada is used as the embedding model. To retrieve the most relevant documents, we've chosen cosine similarity as our unit of measurement.

II. Related Works

A chatbot, by definition, is an application/computer program that has the ability to interact with humans and simulate conversations [10]. Apart from that, chatbots are also harnessed to assist humans in other sectors. This includes business, health, and academia. The first chatbot to ever come into existence was ELIZA in 1966, albeit it had shortcomings in multiple areas. ELIZA was a rigid chatbot, and it deliberately failed to understand the context. This is primarily caused by how ELIZA works, it utilizes pattern matching and a response selection scheme based on templates [11].

In the context of using chatbots to serve information regarding academics, [12] used Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA) to answer questions related to university information. The method used AIML, a rule-based method, to respond to general questions and greetings, whereas LSA provides service-based answers. AIML and other pattern-based methods would still be used in the following years [2], [13], [14], [15].

Named Entity Recognition (NER) tasks were employed in various instances with different algorithms [16], [17]. In essence, NER aims to detect specific entities and extract them. NER works with other mechanisms to generate an answer. In 2018, [17] demonstrated how chatbots take advantage of NER. His method works by first determining what the intent of the user is, then based on that, context info is extracted. When a user is asking about the courses in a major, the first stage is recognizing the intent (asking about a major), then it would seek the entity (what the major is). This entire pipeline was supported by Dialogflow, a cloud-based service.

Cloud-based services during that time, such as IBM Watson Conversation, Dialogflow, and Amazon Lex, are alternatives that were harnessed in the development of chatbots [18], [19], [20], [21], [22], [23]. All of these methods provide the user an interface to create the intent detection and context extraction pipeline like [17] did. But a major advantage these types of services provide is significant convenience, requiring only the preparation of training data.

In the following years, many researchers adopted machine learning techniques. Similarity-based techniques, where the model sought to identify the word most closely related to the user's query (based on certain distance measures), were identified [24]. The dataset of questions and answers was created using supervised approach. The data are then subsequently transformed into numerical vectors used multiple similarity measurements. The best result was cosine similarity.

Machine learning methods, such as K-Nearest Neighbors (KNN), were also detected [25]. KNN was the primary way of classifying the user's question type, which then returned a result based on that question. The final model achieved a 55% accuracy in answering the questions.

Moreover, the development of chatbots for various universities leveraged deep learning techniques [26], [27]. A QA system has been developed for Telkom University that eases access to information regarding students. The approach that was done was a sequence-to-sequence (Seq2Seq) model. A Seq2Seq model consists of two parts, an encoder and a decoder, the former encodes the text (question) into a context vector, whilst the latter decodes the context vector to a corresponding answer. The decoder and encoder architecture may vary, in this case and takes advantage of a two-layered Long Short-Term Memory (LSTM) with an attention mechanism. The result was a robust model that could perceive and understand the queries asked by the user, even when the user uses sub words. The chatbots acknowledges effectiveness in handling the Indonesian language, particularly considering that the language is abundant in sub-words. However, builds the LSTM model needs outsource computing power for the model to work [27]. All the methods mentioned only take advantage of parametric methods, and those ways miss more when faced with answering fixed answers.

III. Material and Methodology

In this study, we suggest using OpenAI GPT-3.5 as the primary Large Language Model (LLM) for generating answers. The retriever component is built around OpenAI Ada as the embedding model. The cosine similarity is utilized as the method for identifying the most pertinent documents during the search process.

In material, there are three types of datasets referred to in this study: a corpus of documents related to academic information and two evaluation datasets for evaluating the retriever and LLM. 111 Documents in the corpus were collected from faculty documents and websites. The evaluation dataset for the retriever consists of 90 questions/queries along with document IDs in the corpus containing the answers to the questions. Each question has a varying number of relevant documents, ranging from 1-2 documents. The evaluation dataset for LLM consists of 60 questions related to academic information and one introductory question, each with a pre-determined answer. This dataset was collected and validated by students and lecturers. The corpus is stored in the ChromaDB vector database by preprocessing each document into chunks based on a set number of tokens, with 1000 tokens per chunk. Considering the context size of >16,385 tokens for each LLM used in this study, several chunks will be sufficient to be loaded as LLM input.

The implementation of the Retrieval-Augmented Generation (RAG) process in chatbot development relies on LangChain's Framework. The framework is integrated with other essential components related to chatbot creation. LangChain is an open-source framework that facilitates the development of LLM-based applications, such as chatbots, translation machines, document summarization, and more.

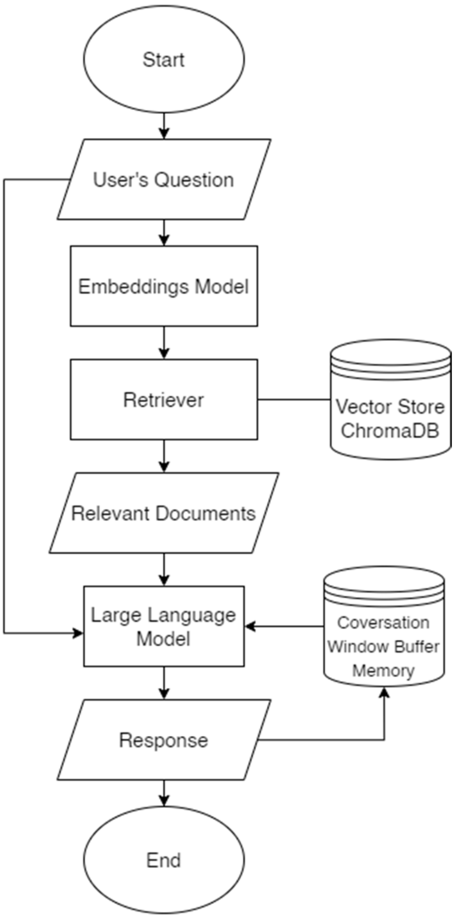

Fig. 1. Flow Chart of Retrieval-Augmented Generation with Conversation Buffer Window Memory

In Fig.1, the mechanisms of RAG have two crucial processes: The Retrieval Process and the Answer Generation Process. In the retrieval process, the retriever must find the most relevant documents to the user's query from a database that contains vector representations of all knowledge documents. In this context, the key components in the retrieval process include vector embeddings, search methods, and the vector database. The quality of embeddings significantly influences the retrieved documents because if the embedding model fails to capture the semantic meaning of the documents, any search method used will not be optimal. Additionally, the choice of search method also impacts the selection of documents by the retriever. Common search methods like similarity search and Maximum Marginal Relevance (MMR) Search will be compared in this research.

Similarity search's aim is to find the most relevant documents to a user's question. These documents are represented by vector embeddings in a high-dimensional space. Therefore, to determine the similarity between embeddings, a suitable metric used is cosine similarity when compared to other metrics such as L2 (Euclidean Distance) and Inner Product.

$$cos(x,y) = \frac{x \cdot y}{||x|\cdot||y|||} \quad (1)$$

Eq.1 is cosine similarity measurement where $x \cdot y$ is the inner product between vectors x and y, and $||x||$ and $||y||$ are the L2 norms or Euclidean lengths of vectors x and y, respectively. This formula measures the similarity in direction between two vectors in high-dimensional space.

A. Maximum Marginal Relevance (MMR) Search

MMR search optimizes not only the similarity to the query but also the diversity among the selected documents. This approach calculates relevance and diversity independently and combines them linearly to produce a metric value called "marginal relevance." A document has a high marginal relevance value if it is relevant to the query and has minimal similarity to the already selected documents. The calculation of MMR is Eq.2.

$$MMR(S_i, R, D) = (1-\lambda).\, sim\ (S_i, R) - \lambda.\, max_{S_j \epsilon D} sim(S_i, S_j) \quad (2)$$

Where S_i is the evaluated search result, R is the search query, and D is the set of search results. Meanwhile, λ is a configurable parameter to control how much emphasis is given to diversity.

By incorporating diversity into the score calculation, MMR tends to return more diverse documents, providing a more comprehensive or varied perspective. Additionally, MMR can address result redundancy by assigning lower marginal relevance scores to documents that have high similarity with other equally relevant documents. The flexibility of the parameter λ in MMR allows us to control the emphasis on diversity.

B. Answer Generation Process

The answer Generation Process is to find a way to generate answers. The Answer Generation begins with feeding the relevant documents and the query into the Large Language Model (LLM) for answer generation. There are several considerations in choosing between OpenAI GPT-3.5 and Google Gemini as a comparison for this purpose.

OpenAI GPT-3.5 [28] is a closed-source LLM, meaning it can only be accessed through the API provided by OpenAI. The costs associated with it are based on the number of tokens used, including both prompt tokens and completion tokens. Closed-source LLMs offer the advantage of not requiring as much computational power as open-source LLMs. Only internet connection is needed. However expenses are both manageable and reasonable if used sporadically.

A Generative Pre-trained Transformer is a natural language model that uses the Transformer algorithm [29]. The Transformer itself is a model architecture that introduced the attention mechanism, a mechanism that allows the model to dynamically focus on specific parts of the input during the processing. The Transformer architecture consists of two main components: an encoder and a decoder. The encoder is responsible for processing the input and creating its contextual representation, while the decoder is used to generate output based on that representation. However, GPT adopts a unique approach by only using the decoder part of the Transformer architecture. By focusing on the decoder, GPT can effectively predict the next word in a sequence context to generate coherent and contextual text.

As a comparison to GPT-3.5, we also used Google Gemini Pro [30], which is reported to have competitive capabilities. For example, in the MMLU benchmark containing multiple-choice questions in 57 subjects including professionals and academics [31], Gemini Pro scored 79.13% and GPT-3.5 scored 70%. Like GPT, the Gemini model is built on the Transformer decoder, with architectural improvements for optimized training and inference on Google's Tensor Processing Unit. Gemini version 1.0 comes in three different sizes: Ultra, Pro, and Nano. This research uses the Gemini Pro version as it is reported to have competitive capabilities against GPT-3.5.

IV. Results and Discussion

We emphasize that our proposed chatbot with retriever and LLM will be compared with other methods. In this paper, we compare our method with other competitive methods, such as Google Gemini, to provide a fair assessment in contrast to OpenAI GPT-3.5. Additionally, we evaluate OpenAI Ada alongside other embedding models, namely Google Embedding-001 and MPNet-Multilingual [32]. The latter stands out for its multilingual capabilities and being an open-source model [33]. The similarity search method is examined against the MMR search, which not only emphasizes similarity but also diversity in responses. This research demonstrates that our proposed methods consistently yield favorable responses to inquiries related to academic information, such as new student admissions, academic programs, and more.

The evaluation of the retriever can be done by inputting a query into the retriever and then assessing how accurate the returned documents are compared to the actual relevant documents. In the evaluation dataset for the retriever, the query used is a question, and the retriever must predict relevant documents from the 111 documents in the corpus. The retriever originating from each combination of searching methods with embedding models will be measured using recall and precision metrics. Recall is a metric that measures the ability of the retriever to correctly identify and retrieve all relevant documents from the corpus. Recall in Eq.3 is defined as the ratio of the number of relevant documents retrieved to the total number of relevant documents in the corpus.

$$Recall = \frac{Number\ of\ Relevant\ Documents\ Retrieved}{Total\ Number\ of\ Relevant\ Documents\ in\ the\ Corpus} \quad (3)$$

The precision metric in Eq.4 evaluates the total number of documents retrieved by assessing the accuracy of the retrieved documents. It is calculated as the ratio of the number of relevant documents retrieved to the total number of documents retrieved including both relevant and irrelevant ones. Mathematically, precision is defined as:

$$Precision = \frac{Number\ of\ Relevant\ Documents\ Retrieved}{Total\ Number\ of\ Documents\ Retrieved} \quad (4)$$

As previously explained, since each query has a varying number of relevant documents between 1-2, the number of k documents that must be returned by the retriever during the evaluation process depends on the number of relevant documents the related question has in the evaluation dataset. However, in production, the k parameter is always set to 2, considering that having more context in the answer generation process is better than limited context. Additionally, this number is considered efficient in the use of input tokens in the LLM and is deemed sufficient to provide context for the LLM to answer user questions.

The quality of answers by the LLM measure with BLEU (Bilingual Evaluation Understudy) [34] score and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [35] score evaluation metrics. The BLEU score is a precision-oriented metric. It provides a nuanced understanding of the degree to which the words or n-grams generated by the LLM align with reference answers. Beyond single words, BLEU considers n-grams. This facilitates a detailed evaluation of the quality of the generated text. In the context of applying evaluation metrics across various test samples, BLEU scores are averaged. The BLEU score is computed in the Eq.5.

$$BLEU - N = BR \quad \left(\prod_{n=1}^{N} p_n\right)^{\frac{1}{N}} \quad (5)$$

Where the BLEU precision is computed as the geometric mean of the n-gram precisions, the n-gram precision is calculated as the clipped count of n-grams in the output divided by the total count of n-grams in the output.

ROUGE score is another established evaluation metric used in text generation. Unlike the BLEU score which is precision-oriented, the ROUGE score is recall-oriented. This means it measures the extent to which the words or n-grams generated by the model are present in the reference answers. The ROUGE score considers several aspects of the generated text. These include unigram recall (ROUGE-1), bigram recall (ROUGE-2), and longest common subsequence (ROUGE-L). Each of these aspects provides a different perspective on the quality of the generated text. Unigram recall (ROUGE-1) measures the overlap of single words between the generated text and the reference answers. Bigram recall (ROUGE-2) measures the overlap of two consecutive words (bigrams) between the generated text and the reference answers. The longest common subsequence (ROUGE-L) measures the longest sequence of words that appear in both the generated text and the reference answers in the same order. The ROUGE score is computed using Eq.6.

$$ROUGE - N = \frac{\Sigma_{snt' \in C} \Sigma_{n-gram \in snt'} Count_{match}(n-gram)}{\Sigma_{snt' \in C} \Sigma_{n-gram \in snt'} Count(n-gram)} \quad (6)$$

Where snt' is one of the sentences in the references set C. This means every n-gram of the generated answer (candidate) is compared to every single one of the sentences in the reference set.

A. Retrievers Evaluation

In Retrievers Evaluation, the degree of diversity parameter is set for MMR search to 0.5, which is the default value determined by the ChromaDB vector store. The comparison method is show in TABLE 1.

TABLE I. Retrievers Comparison Results

Searching Method	Embeddings Model	Recall	Precision
Cosine Similarity Search	OpenAI Ada (Our Proposed)	**86.07%**	**77.05%**
	MPNet-Multilingual	53.28%	45.90%
	Google Embedding-001	22.95%	59.02%
MMR Search	OpenAI Ada	59.02%	48.36%
	MPNet-Multilingual	47.54%	39.34%
	Google Embedding-001	22.95%	18.85%

In TABLE 1, the evaluation results of the retriever show a comparison between cosine similarity search using the OpenAI Ada embedding model and the MMR search method with other embedding models. It is observed that by using the parameter for the number of returned documents $k = 2$, our proposed retriever has the best recall and precision values among other methods. This result aligns with our priority in building this chatbot, where it is better to have an excess of context (recall-oriented) in the answer-generation process.

The OpenAI Ada model is the best model for capturing semantic meaning in the text compared to MPNet-Multilingual and Google Embedding-001. This is evident in both types of search methods, where OpenAI Ada excels in both recall and precision. Moreover, when conducting a comparison of the OpenAI Ada model with other search methods, the cosine similarity gives better results. The degree of diversity parameter is not suitable as it fails to deliver diverse and relevant documents, and instead, it presents irrelevant ones. Therefore, fine-tuning the degree of diversity parameter is necessary. Grid search explores a range from 0.3 to 0.8 with a step size of 0.05. The optimal outcomes obtained from this fine-tuning process are presented in TABLE II.

TABLE II. FINE TUNING DEGREE OF DIVERSITY PARAMETER

Retriever	Degree of Diversity	Recall	Precision
MMR Search on OpenAI Ada (Before Fine-tuning)	0.5	59.02%	48.36%
MMR Search on OpenAI Ada (After Fine-tuning)	0.75	77.86%	67.21%
Cosine Similarity on OpenAI Ada (Proposed Retriever)	-	**86.07%**	**77.05%**

Thus, it can be concluded that MMR Search has good potential if the degree of diversity parameter is fine-tuned beforehand. However, these results have limitations, as they are only valid for the number of retrieved documents (k) set to 2 by the retriever. The findings may not be interpreted similarly if k is set to a larger value, as there is a possibility that Cosine Similarity may return redundant documents when k is large.

TABLE III. SETTING THE RETRIEVED DOCUMENTS PARAMETER K=1

Metrics Evaluation	OpenAI Ada	Google Embedding-001	MPNet-Multilingual
Accuracy	85.25%	6.56%	65.57%

In Table III, the results attempted to change k to 1, considering that some questions in the dataset can be answered with just one context. However, $k = 2$ provides more comprehensive information to the Language Model (LLM). When k is reduced to 1, the focus shifts to whether the returned document is relevant or not. In this case, evaluating accuracy alone is sufficient as it measures how well the model correctly identifies the relevance of a single document. TABLE III assesses cosine similarity search because the idea of MMR Search is not relevant when only one document is returned since one document is not diverse.

The evaluation results indicate that the use of OpenAI Ada Embeddings shows consistency in the changes of the k values. On the contrary, Google Embedding-001 significantly lags behind both other embedding models. The utilization of the MPNet-Multilingual embedding model in experiments with $k = 1$ resulted in a notable 23.06% increase when compared to the prior experiment with $k = 2$. This finding suggests that MPNet-Multilingual may serve as a cost-effective alternative to OpenAI Ada.

B. LLMs Evaluation

The chatbot uses a Conversation Window Buffer Memory to remember past conversations with people and respond better to follow-up questions. The System Message is utilized to control the behavior of the Language Model (LLM), ensuring that the responses provided are not textbook-like. Additionally, to obtain precise responses from the LLM, the temperature parameter is set to 0. TABLE IV shows the experiment for comparing methods in the LLM.

TABLE IV. EVALUATION SCORE IN COMPARISON LLM METHODS

LLM	Rouge1	Rouge2	RougeL	RougeLsum	BLEU Score
Gemini Pro	23.31%	12.19%	20.24%	20.24%	5.65%
GPT-3.5-Turbo (Our proposed)	**47.24%**	**34.85%**	**43.34%**	**43.37%**	**25.14%**

In evaluating language models, GPT-3.5-Turbo consistently surpasses Gemini Pro across multiple linguistic metrics. Notably, Rouge1, which gauges unigram overlap, indicates a marked proficiency of GPT-3.5-Turbo in capturing individual word matches, as evidenced by a substantially elevated score relative to Gemini Pro. This proficiency extends to Rouge2, assessing bigram overlap, wherein GPT-3.5-Turbo showcases superior contextual comprehension, thereby outperforming Gemini Pro. Furthermore, both RougeL and RougeLsum, emphasizing the longest common substring, underscore GPT-3.5-Turbo's prowess in generating more coherent and extensive substrings when contrasted with Gemini Pro.

In the BLEU Score evaluation, GPT-3.5-Turbo outperforms Gemini Pro, signifying better linguistic quality. The higher BLEU score reflects improved overlap with reference answers' n-grams, highlighting GPT-3.5-Turbo's superior text generation. Overall, GPT-3.5-Turbo emerges as a superior choice, aligning with ROUGE and BLEU metrics. Its ability to produce similar, high-quality text reinforces its efficacy in supporting chatbot responses for academic information, solidifying its role among Large Language Models.

V. CONCLUSION

This paper presents a comprehensive study on the development and evaluation of a chatbot for academic information utilizing the Retrieval-Augmented Generation (RAG) technique, specifically integrating OpenAI GPT-3.5 as the Large Language Model (LLM). The research explores the performance of the proposed chatbot against competitive methods such as Google Gemini, considering both retriever and language model aspects.

In the retriever evaluation, cosine similarity search and OpenAI Ada embeddings outperform other methods in terms of recall and precision, emphasizing the effectiveness of the proposed approach in providing relevant context for answer generation. Fine-tuning the degree of diversity parameter in MMR Search reveals its potential to compete with the proposed retriever, emphasizing the importance of parameter optimization. Further investigation into the number of retrieved documents (k) reveals that, with $k = 1$, OpenAI Ada maintains consistency in recall and precision, outperforming other embedding models. Notably, MPNet-Multilingual emerges as a cost-effective alternative to OpenAI Ada, showing an increase in performance with $k = 1$, suggesting its viability for local use without internet dependence.

The evaluation of Large Language Models (LLMs) using ROUGE and BLEU scores demonstrates the superiority of GPT-3.5-Turbo over Gemini Pro in terms of unigram and bigram overlap, as well as overall text quality. GPT-3.5-Turbo consistently generates text that aligns better with reference answers, reinforcing its suitability for chatbot responses related to academic information. The effectiveness of the chatbot's responses is heavily influenced by the quality of the knowledge it possesses. Therefore, exploring and expanding the knowledge base by adding relevant documents to the vector database could significantly enhance the chatbot. Additionally, a thorough analysis of existing open-source Language Model (LLM) implementations is warranted to identify alternative options and provide researchers with a broader choice for developing academic chatbots.

Acknowledgment

We would like to express our gratitude to Universitas Airlangga (1880/UN3.FTMM/PT.01.03/2023).

References

[1] R. Hardi, A. Naim, Muhammad, V. A. Pitogo, Agung Sakti Pribadi, and Jack Febrian Rusdi, "Academic Smart Chatbot to Support Emerging Artificial Intelligence Conversation," 2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA), Nov. 2022,

[2] S. N. M. S. Pi and M. A. Majid, "Components of Smart Chatbot Academic Model for a University Website," *IEEE Xplore*, Dec. 01, 2020.

[3] Thikraa Mohammed Alharethi, "Autoresponder using Chatbot for Educational Services," 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jan. 2023.

[4] A. Chaidrata *et al.*, "Intent Matching based Customer Services Chatbot with Natural Language Understanding," *2021 5th International Conference on Communication and Information Systems (ICCIS)*, Oct. 2021, doi: https://doi.org/10.1109/iccis53528.2021.9646029.

[5] M. Y. Helmi Setyawan, R. M. Awangga, and S. R. Efendi, "Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot," *IEEE Xplore*, Oct. 01, 2018.

[6] Yuanxun Ethan Wang, Winata Liadylova Putera, H. Lucky, and Andry Chowanda, "Chatbot Application to Automate Services in FnB Business Using Seq2Seq LSTM," *2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, Nov. 2022, doi: https://doi.org/10.1109/icimcis56303.2022.10017854.

[7] Pariwat Maktapwong, Pichathat Siriphornphokha, Supawadee Tubglam, and Aurawan Imsombut, "Message Classification for Breast Cancer Chatbot using Bidirectional LSTM," *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, Jul. 2022, doi: https://doi.org/10.1109/itc-cscc55581.2022.9895035.

[8] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems, 2020.

[9] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," arXiv (Cornell University), Sep. 2023, doi: https://doi.org/10.48550/arxiv.2309.15217.

[10] Rachana Vannala, S.B. Swathi, and Yuvaraj Puranam, "AI Chatbot For Answering FAQ's," *2022 IEEE 2nd International Conference on Sustainable Energy and Future Electric Transportation (SeFeT)*, Aug. 2022, doi: https://doi.org/10.1109/sefet55524.2022.9908774.

[11] B. Morris, "The components of the Wired Spanning Forest are recurrent," *Probability Theory and Related Fields*, vol. 125, no. 2, pp. 259–265, Feb. 2003, doi: https://doi.org/10.1007/s00440-002-0236-0.

[12] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2017, doi: https://doi.org/10.1109/icacci.2017.8126057.

[13] L. Tommy, C. Kirana, and L. Riska, "The Combination of Natural Language Processing and Entity Extraction for Academic Chatbot," *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, Oct. 2020, doi: https://doi.org/10.1109/citsm50537.2020.9268851.

[14] Reoof Al-Jedaie, Reem Al-Hindy, Hanan Al-Onazi, Elham Kariri, and Fatma Masmoudi, "A chatbot for Academic advising," International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Nov. 2022.

[15] D. Sebastian and Kristian Adi Nugraha, "Academic Customer Service Chatbot Development using TelegramBot API," 2021 2nd International Conference on Innovative and Creative Information Technology (ICITech), Sep. 2021.

[16] Tarek Ait Baha, Mohamed El Hajji, Youssef Es-Saady, and Hammou Fadili, "The impact of educational chatbot on student learning experience," *Education and Information Technologies*, Sep. 2023, doi: https://doi.org/10.1007/s10639-023-12166-w.

[17] H. T. Hien, P.-N. Cuong, L. N. H. Nam, H. L. T. K. Nhung, and L. D. Thang, "Intelligent Assistants in Higher-Education Environments," *Proceedings of the Ninth International Symposium on Information and Communication Technology - SoICT 2018*, 2018, doi: https://doi.org/10.1145/3287921.32879

[18] H. Agus Santoso *et al.*, "Dinus Intelligent Assistance (DINA) Chatbot for University Admission Services," *2018 International Seminar on Application for Technology of Information and Communication*, Sep. 2018, doi: https://doi.org/10.1109/isemantic.2018.8549797.

[19] C. Chun Ho, H. L. Lee, W. K. Lo, and K. F. A. Lui, "Developing a Chatbot for College Student Programme Advisement," IEEE Xplore, Jul. 01, 2018.

[20] J. Heo and J. Lee, "CiSA: An Inclusive Chatbot Service for International Students and Academics," *HCI International 2019 – Late Breaking Papers*, pp. 153–167, 2019, doi: https://doi.org/10.1007/978-3-030-30033-3_12.

[21] R. Carrizales, J. Ramirez, J. Armas-Aguirre, and E. E. Grandon, "Cognitive services to improve user experience in searching for academic information based on chatbot," 2019 IEEE XXVI International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 2019, pp. 1-4, doi: 10.1109/INTERCON.2019.8853572.

[22] V. R, B. B, A. S, and D. M, "AI Based Student Bot for Academic Information System using Machine Learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 590–596, Mar. 2019, doi: https://doi.org/10.32628/cseit1952171.

[23] A. Alkhoori, M. A. Kuhail and A. Alkhoori, "UniBud: A Virtual Academic Adviser," 2020 12th Annual Undergraduate Research Conference on Applied Computing (URC), Dubai, United Arab Emirates, 2020, pp. 1-4, doi: 10.1109/URC49805.2020.9099191.

[24] A. Verma *et al.*, "University Chatbot System using NLP," SSRN Electronic Journal, 2022, doi: https://doi.org/10.2139/ssrn.4255753.

[25] M. Ula, R. Hardi, and I. Hipiny, "An Improved Structure for Academic Information Services through AI Chatbots," *Journal of Engineering Science and Technology Review*, vol. 16, no. 5, pp. 164–173, 2023, doi: https://doi.org/10.25103/jestr.165.20.

[26] Alaa Aloqayli and Hoda Ahmed Abdelhafez, "Intelligent Chatbot for Admission in Higher Education," *International Journal of Information and Education Technology*, vol. 13, no. 9, pp. 1348–1357, Jan. 2023, doi: https://doi.org/10.18178/ijiet.2023.13.9.1937.

[27] Y. W. Chandra and S. Suyanto, "Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model," *Procedia Computer Science*, vol. 157, pp. 367–374, 2019, doi: https://doi.org/10.1016/j.procs.2019.08.179.

[28] OpenAI, "GPT-4 Technical Report," *arXiv (Cornell University)*, Mar. 2023, doi: https://doi.org/10.48550/arxiv.2303.08774.

[29] A. Vaswani *et al.*, "Attention Is All You Need," *Advances in neural information processing systems*, Dec. 05, 2017.

[30] Gemini Team, "Gemini: A Family of Highly Capable Multimodal Models," arXiv preprint arXiv:2312.11805, 2023.

[31] D. Hendrycks *et al.*, "Measuring Massive Multitask Language Understanding," *arXiv (Cornell University)*, Sep. 2020, doi: https://doi.org/10.48550/arxiv.2009.03300.

[32] N. Reimers and I. Gurevych, "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation," *arXiv:2004.09813 [cs]*, Oct. 2020.

[33] Niklas Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: Massive Text Embedding Benchmark," *arXiv (Cornell University)*, Oct. 2022, doi: https://doi.org/10.48550/arxiv.2210.07316.

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02, 2001, doi: https://doi.org/10.3115/1073083.1073135.

[35] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," ACLWeb, Jul. 01, 2004.