Volume 11 Issue 1 Year 2026 | 298-309 ISSN: 2527-9866
Received: ***13-01-2026*** | *Revised:* ***20-01-2026*** | *Accepted:* ***12-02-2026***

Implementation of Retrieval-Augmented Generation Method on Large Language Model for Development of Campus Service and Information Chatbot

Muhammad Dzaki Salman[1], Rahmaddeni[2], Torkis Nasution[3], Susanti[4]
[1,2,3,4] Indonesian University of Science and Technology (USTI), Pekanbaru, Indonesia, 28299
e-mail: muhammaddzakisalman@gmail.com[1], rahmaddeni@usti.ac.id[2], torkisnasution@usti.ac.id[3], susanti@usti.ac.id[4]
*Correspondence: muhammaddzakisalman@gmail.com

Abstract: *Large Language Models have the potential to improve the quality of information services in higher education environments through responsive and natural interactions. However, LLMs are prone to generating answers that are not supported by valid knowledge sources due to knowledge cut-off limitations. This study implements Retrieval-Augmented Generation on LLMs to build an information service chatbot for the Indonesian University of Science and Technology (USTI). RAG is built using a hybrid retrieval mechanism that combines dense retrieval and sparse retrieval (BM25) through Reciprocal Rank Fusion, and is equipped with cross-encoder reranking. The knowledge base is compiled from official and public documents obtained through the USTI website. The evaluation was conducted using 13 test queries by comparing several configurations to analyze the contribution of each component. The evaluation results show that the hybrid retrieval configuration produces the best retrieval performance with Precision@3 of 71.7%, Recall@3 of 87.5%, and NDCG@3 of 96.3%. In addition, the application of RAG improved the quality of answers compared to LLM without retrieval, as shown by an increase in BERTScore-F1 from 84.8% to 89.4% and a faithfulness score of 88.8%. These findings indicate that RAG integration improves the relevance of LLM answers to source documents, with the hybrid configuration providing an optimal balance between retrieval quality and faithfulness.*

Keywords: *Retrieval-Augmented Generation, Large Language Model, hybrid retrieval, chatbot*

1. INTRODUCTION

The development of digital technology has brought significant transformations in the provision of information services in various sectors, including higher education. Higher education institutions are required to provide information systems that are responsive, accurate, and easily accessible to the academic community and the general public. One solution that is currently growing rapidly is chatbots based on Large Language Models (LLMs) that are capable of processing questions and providing quick and automatic responses. LLMs are artificial intelligence models based on neural networks that are trained using large amounts of text data from various sources so that they are able to understand complex language patterns and produce coherent text that resembles human writing [1].

The development of LLMs such as GPT and BERT has improved chatbots' ability to interact naturally compared to rule-based approaches. However, LLMs have a major limitation in the form of knowledge cut-off, as they are trained on data corpora up to a certain period, so they are not always able to provide the latest information [2], [3]. This limitation can trigger hallucination, a condition where the model generates linguistically coherent answers that are not supported by valid facts. This problem can be overcome by applying Retrieval-Augmented

Generation (RAG) to Large Language Models. RAG is a method that combines the power of generative models with the knowledge search process from external databases, so that it can produce answers that are more factual and traceable [4].

Previous studies have reported that the implementation of Retrieval-Augmented Generation (RAG) is effective in improving the performance of LLM-based question-answering systems, particularly in information services [5]. In the context of higher education, RAG-based academic chatbots utilize vector databases and generative models to generate answers based on relevant documents, demonstrating the effectiveness of RAG as a grounded answering approach in institutional information scenarios [6]. In this approach, chatbots limit their answers to information available in the given context or source [7]. Another study that implemented RAG in a Telegram chatbot for a student information system also reported a decrease in the hallucination rate and an increase in user satisfaction with the accuracy and relevance of responses [8]. These findings indicate that integrating document retrieval mechanisms into the generative process of LLM through RAG consistently produces reliable, responsive, and relevant systems for providing information, particularly in knowledge-intensive higher education environments.

This study implements the RAG method on a LLMs to build a service and information chatbot for the Indonesian USTI with a hybrid retrieval approach that combines vector-based and keyword-based searches, as well as Reciprocal Rank Fusion and cross-encoder reranking mechanisms to improve context selection quality. This research focuses on answering the following questions.

1. To what extent does the implementation of RAG improve the grounding and faithfulness (attachment) of LLM responses to source documents compared to LLMs without retrieval?

2. How do the respective components in the retrieval pipeline, including vector retrieval, keyword retrieval, Reciprocal Rank Fusion (RRF), and cross-encoder reranking, contribute to improving the quality of relevant document selection and ranking?

3. To what extent can the hybrid retrieval approach improve document search quality compared to a single retrieval approach?

Thus, this study is expected to contribute to evaluating the effectiveness of hybrid retrieval-based RAG in campus information service chatbots and analyzing the technical contribution of each retrieval component to improving the context and faithfulness of LLM responses.

2. METHOD

A. Research Flow

This study was designed and conducted systematically to ensure that each stage was in line with the predetermined objectives. The series of research stages is shown in Figure 1.

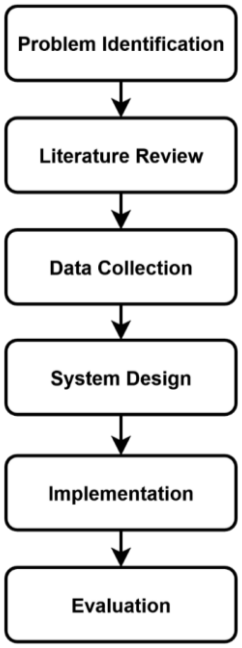


Figure 1. Research Flow

1. Problem Identification

Academic information at the USTI is scattered across various official campus sources, making it difficult for students and the general public to obtain specific information quickly and efficiently. The use of LLMs has the potential to improve the quality of information services through responsive and natural interactions. However, the use of LLM without a grounding mechanism against verified institutional documents risks producing answers that are inconsistent with official campus information. Therefore, an approach is needed that can integrate the generative capabilities of LLM with valid campus knowledge sources so that chatbots can provide accurate, relevant, and accountable information.

2. Literature Review

Literature study was done to gain a theoretical and empirical foundation related to the topic of the research. This phase involved the examination of existing studies on Large Language Models, information service chatbots, information retrieval techniques, and the application of Retrieval-Augmented Generation. The literature study seeks to explore the approaches that have been employed, their strengths and weaknesses, and the research gaps that can be developed further in relation to campus information services.

3. Data Collection

Data gathering was done by compiling official documents on campus that served as sources of knowledge for the chatbot system. All documents used were public and did not contain sensitive personal information. Documents gathered included academic guidelines, institutional regulations, administrative service information, and other supporting documents. All data was in text form and contained information that was valid and relevant to the needs of chatbot users.

4. System Design

The system design incorporates a RAG model that leverages information retrieval and the generation capabilities of LLMs. The system incorporates an indexing part for constructing the external knowledge base and a RAG part for document retrieval, context construction, and answer generation, in addition to a web interface for user interaction. The system flow is depicted in Figure 2.

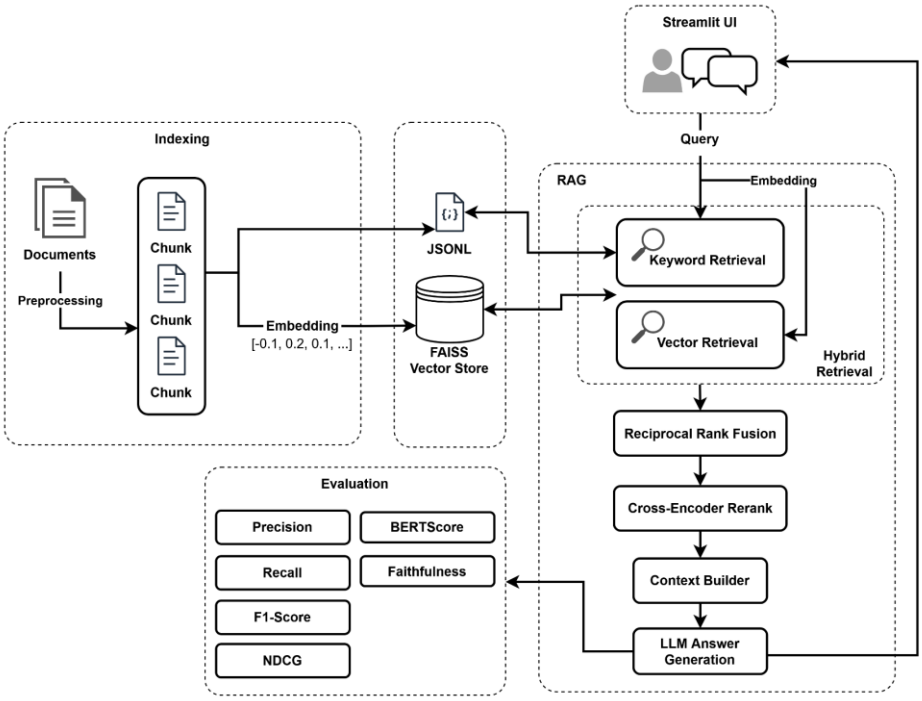


Figure 2. System Flow

a. Indexing

In the RAG pipeline, the process of indexing is intended to create an external knowledge graph that can aid the LLM response generation process. This process involves the transformation of raw documents into a form that can be searched based on semantic similarity and keyword matching [9]. The indexing process has two primary steps, which are chunking and embedding. The output of this process is stored in FAISS as a vector store, which will be used in the semantic search process [10][11]. The output of chunking is also stored in JSONL form for keyword searching [12].

1. Data Preprocessing

The preprocessing step includes Unicode normalization based on the Normalization Form Compatibility Composition (NFKC) standard, non-standard typography to standard ASCII character conversion, and whitespace normalization by reducing spaces and double lines. This is done to remove unnecessary variation in tokens and maintain the quality of semantic representation.

2. Chunking

The chunking stage is the first step in the RAG pipeline that decomposes the long text documents into smaller units of information that can be processed by LLMs based on the limitations of the context window. The formation of smaller units of information helps in the retrieval of correct information, in addition to helping the embeddings in understanding the semantic meaning of the information in a more focused way, ensuring that the alignment of the query and document in the retrieval process is more effective [13]. The chunks are created based on the token limit or the natural structure of the documents, such as paragraphs and sentences [14].

3. Embedding

Tokenization aims to maintain input structure consistency and compatibility with the encoder architecture. Input text is converted into a sequence of numeric tokens using a WordPiece-based tokenizer with the following formula [15].

1) Tokenization

Tokenization aims to maintain input structure consistency and compatibility with the encoder architecture. Input text is converted into a sequence of numeric tokens using a WordPiece-based tokenizer with the following formula.

$$X = Tokenizer(T) = [CLS], w_1, w_1, \ldots w_n, [SEP]$$

Special tokens [CLS] and [SEP] are used as start and end markers, as applied in BERT-based models.

2) Transformer Encoding

The tokens that have been converted into IDs are sent to the transformer encoder, which uses a self-attention mechanism with the following formula.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

This mechanism allows the model to capture semantic dependencies between tokens contextually. The output of this stage is a representation vector for each token.

3) Mean Pooling

To obtain a representative vector of the entire text, mean pooling is performed on the hidden state of all tokens (except padding) using the following formula.

$$\vec{e} = \frac{1}{n}\sum_{i=1}^{n} h_i$$

Where h_i is the hidden state output for the i token, and n is the actual number of tokens.

4) L2 Normalization

In order for the embedding to have a uniform scale and be usable for cosine similarity calculations, L2 normalization is performed using the following formula.

$$\hat{e} = \frac{\vec{e}}{\|\vec{e}\|_2} = \frac{\vec{e}}{\sqrt{\sum_{i=1}^{d} e_i^2}}$$

The final result $\hat{e}$ is a fixed-dimensional unit vector ready to be stored in the FAISS vector store.

b. Retrieval-Augmented Generation

RAG runs the entire retrieval and answer generation process when a query is made to the system. RAG consists of the following stages.

1. Query

The query received is forwarded to the retrieval system to search for relevant answers from the existing collection of documents or data.

2. Hybrid Retrieval

At this stage, document retrieval is performed using a hybrid retrieval approach, namely vector retrieval using the intfloat/multilingual-e5-base model and keyword retrieval using Best Matching 25 (BM25).

3. Reciprocal Rank Fusion (RRF)

The results from vector and keyword retrieval are combined using the RRF technique to obtain a more robust and balanced list of results. RRF is calculated using the following formula.

$$RRFscore(d \in D) = \sum_{r \in R} \frac{1}{k + rank_r(d)}$$

4. Cross-Encoder Reranking

The cross-encoder reranking stage reorders document or chunk candidates obtained from the RRF process by scoring each query–document pair to better reflect their semantic relevance.

5. Context Builder

The context builder takes the reranked results and assembles them into a structured form for the LLM by choosing a few top-ranked documents, retaining key metadata, and formatting the text for optimal context use.

6. Answer Generation

LLM generates answers based on the context of the documents it has been trained on. The model used is LLaMA 3.1 8B Instant, which is run through Groq's high-performance inference engine.

5. Implementation

The system implementation was carried out using the Python programming language. Knowledge documents were processed through a chunking mechanism, represented as vector embeddings using the intfloat/multilingual-e5-base model, and stored in FAISS as a vector store. The retrieval process applies a hybrid approach that combines vector-based semantic search and keyword search using BM25, with the results combined through Reciprocal Rank Fusion (RRF) and re-ranked using cross-encoder/mmarco-mMiniLMv2-L12-H384-v1. The selected context is then used as a prompt for the LLaMA 3.1 8B Instant generative model via the Groq API to generate answers.

6. Evaluation

a. Precision

Precision measures how accurately the system retrieves relevant results from the total number of results retrieved. The formula used is as follows.

$$Precision = \frac{TP}{TP + FP}$$

where True Positive (TP) is the number of relevant data successfully retrieved, and False Positive (FP) is the number of irrelevant data also retrieved.

b. Recall

Recall measures the completeness of the system in finding all relevant data from the entire available data. Recall is calculated using the following formula.

$$Recall = \frac{TP}{TP + FN}$$

False Negative (FN) as the number of relevant data that was not successfully retrieved.

c. F1-Score
F1-Score is the average of precision and recall, reflecting the balance between the two, calculated using the following formula.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

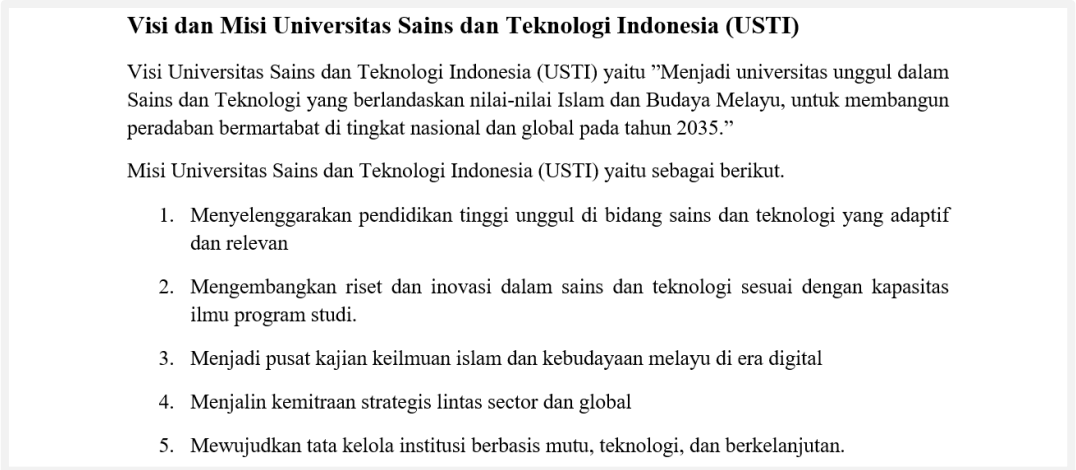
d. BERTScore
BERTScore is a semantic-based evaluation metric that calculates the similarity between the generated and reference texts based on the cosine similarity between the embeddings of the tokens in a BERT model, generating precision, recall, and F1 scores as semantic similarity measures [16].

e. Normalized Discounted Cumulative Gain (NDCG)
NDCG is an evaluation metric for ranking, which judges the quality of ranked results of the retrieval task in terms of relevance, where the relevant items at lower ranks are discounted based on position [17].

f. Faithfulness
Faithfulness evaluates whether the answers generated by the language model are fully supported by the retrieved document context, ensuring that claims in the response can be traced back to the provided evidence and do not introduce unsupported information [18].

3. RESULTS AND DISCUSSION

A. Dataset
The dataset is created from a set of documents that are extracted from the official USTI website, where each document holds one or more specific factual information. The dataset consists of 29 text documents that are categorized into eight main categories, which are institutional profiles, institutional history, academic services, campus facilities, student admissions, campus life, cooperation and partners, and other supporting documents. An example of the dataset is shown in Figure 3.



Visi dan Misi Universitas Sains dan Teknologi Indonesia (USTI)

Visi Universitas Sains dan Teknologi Indonesia (USTI) yaitu ”Menjadi universitas unggul dalam Sains dan Teknologi yang berlandaskan nilai-nilai Islam dan Budaya Melayu, untuk membangun peradaban bermartabat di tingkat nasional dan global pada tahun 2035.”

Misi Universitas Sains dan Teknologi Indonesia (USTI) yaitu sebagai berikut.

1. Menyelenggarakan pendidikan tinggi unggul di bidang sains dan teknologi yang adaptif dan relevan
2. Mengembangkan riset dan inovasi dalam sains dan teknologi sesuai dengan kapasitas ilmu program studi.
3. Menjadi pusat kajian keilmuan islam dan kebudayaan melayu di era digital
4. Menjalin kemitraan strategis lintas sector dan global
5. Mewujudkan tata kelola institusi berbasis mutu, teknologi, dan berkelanjutan.

Figure 3. Dataset

B. Indexing

The document is divided into various chunks based on an adaptive chunking word splitting strategy, which involves calculating the distribution of the total number of words in the document, then calculating the adaptive chunk size divided by an adjustment factor. The chunking parameters are shown in Table .

Table 1. Chunking Parameter Configuration

Parameter	Value	Description
Target Chunk Size	150 words	Default target size per chunk
Minimum Chunk Size	180 words	Minimum limit to prevent chunks from being too short
Maximum Chunk Size	420 words	Maximum limit to prevent chunks from being too long
Chunk Overlap	40 words	Number of words overlapping between chunks
Overlap Ratio	25%	Minimum overlap ratio relative to chunk size
Adaptive Percentile	75	Word distribution percentile used to determine chunk size
Adaptive Divisor	3	Divisor used to adjust the adaptively calculated chunk size

Chunks are stored in a structured format and include metadata such as document source, chunk index, and word range. The chunk results can be seen in Figure 4.


```
{
    "id": "01a-profil-identitas-resmi-usti-0",
    "content": "Universitas Sains dan Teknologi Indonesia (USTI)  ... Sumber: https://usti.ac.id",
    "metadata": {
        "source": "01a_Profil_Identitas_Resmi_USTI.pdf",
        "slug": "01a-profil-identitas-resmi-usti",
        "chunk_index": 0,
        "word_range": [0, 95]
        }
}
```


Figure 4. Chunk Result

Each chunk is then converted into a fixed-dimensional embedding vector, where a single chunk of text is represented as a series of numerical values that capture the semantic meaning of the text content. An example of the embedding results is shown in Table .

Table 2. Embedding Result

Chunk	Embedding Result
Indonesian University of Science and Technology (USTI) berlokasi di Jl. Purwodadi	[0] -0.70615008e-03 [1] 0.65676354e-02 [2] ... [767] 0.05799712

C. Retrieval

The configuration of the hybrid retrieval mechanism is shown in Table .

Table 3. Retrieval Configuration

Method	Parameter	Value	Description
Vector Retrieval	Top-K Candidates	15	Number of candidate chunks retrieved from dense retrieval
	Top-K Multiplier	3×	Multiplication factor to ensure broader coverage
Keyword Retrieval	Top-K Candidates	10	Number of candidate chunks retrieved using BM25
	Lexical Weight	1.0	Weight assigned to BM25 results during the fusion process

The results of vector and keyword retrieval are combined using Reciprocal Rank Fusion (RRF). The RRF configuration is shown in Table .

Table 4. Reciprocal Rank Fusion Configuration

Parameter	Value	Description
RRF Constant (k)	15	Value used to moderate the influence of high-ranked results
Combined Retriever	FAISS + BM25	Combination of dense and sparse retrieval methods
Lexical Weight	1.0	Additional weight applied to BM25 results

The following candidates for hybrid retrieval are reordered by the cross-encoder cross-encoder/mmarco-mMiniLMv2-L12-H384-v1 model to refine the precision of context selection. At this point, the top 10 candidates from the fusion step are taken as input for the reranker, and the three chunks with the highest scores are chosen as the final context for use in the answer generation step.

D. User Interface

The user interface is built on the web using Streamlit in an interactive conversation view. The interface view can be seen at Figure 5.

Figure 5. User Interface

E. Generation

The generation results show that the system is capable of producing structured, relevant answers that are appropriate to the context of the user's question, utilizing information sourced from internal campus documents. The generation results are shown in Figure 6.

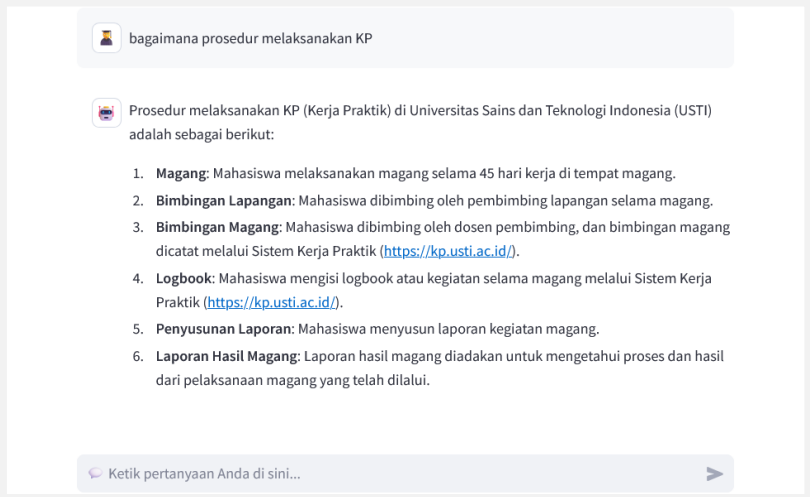


Figure 6. Generation Results

F. Evaluation

The search evaluation was done using 13 test queries that were compiled based on typical questions asked to the campus information service. The relevance of the documents was determined manually by matching the document content with the query intent, with each query linked to one or more relevant documents as ground truth. The relevance annotation was done by a single annotator, which was the author, as shown in Figure 7.


```
{
    "question": "Dimana lokasi kampus utama USTI berada dan bagaimana cara menghubungi universitas untuk layanan informasi publik?",
    "relevant_chunk_ids": ["01b-profil-alamat-dan-kontak-usti-0"],
    "reference_answer": "Jawaban: Kampus utama USTI berada di Jl. Purwodadi Indah ...  / https://usti.ac.id."
}
```


Figure 7. Test Dataset

1. Retrieval Evaluation

Retrieval evaluation is measured using the metrics Precision, Recall, F1-Score, and Normalized Discounted Cumulative Gain (NDCG). The retrieval evaluation results are shown in Table .

Table 5. Retrieval Evaluation Results

Metrics	Score (%)
Precision@3	71,8
Recall@3	87,5
F1-Score@3	78.9
NDCG@3	96.4

The Precision@3 value of 71.8% indicates that most of the chunks retrieved by the system are relevant to the query. The Recall@3 value of 87.50% indicates that the majority of relevant chunks in the knowledge base were successfully found. Meanwhile, the NDCG@3 value of 96.4% indicates excellent ranking quality, where the most relevant chunks tend to be at the top of the list.

2. Comparative Evaluation and Ablation Study

The purpose of this evaluation is to compare the performance of various configurations of RAG and understand the contribution of each component. The results of the comparative evaluation and ablation study are shown in Table .

Table 6. Comparison Results

Metode	Precision@3 (%)	Recall@3 (%)	F1-Score@3 (%)	NDCG@3 (%)
Keyword Retrieval (BM25 + RAG)	69.2	84.3	76.3	94.5
Vector Retrieval (FAISS + RAG)	64.1	78.1	70.4	88.2
Hybrid + RRF + Rerank	71.7	87.5	78.8	96.3

2. Generational Evaluation

Generation evaluation was performed using BERTScore and the faithfulness metric from the RAGAS framework. The generation evaluation results are shown in Table .

Table 7. Generation Evaluation Results

Configuration	Precision (%)	Recall (%)	F1-Score (%)	Faithfulness (%)
LLM-only	84.1	85.5	84.8	-
BM25 + RAG	89.2	89.9	89.5	88.1
FAISS + RAG	89.1	89.5	89.3	93.6
Hybrid (BM25 + FAISS) + RAG	89.1	89.7	89.4	88.8

The BERTScore-F1 value of 89.4% indicates a high degree of semantic similarity between the system's answers and the reference answers. The faithfulness score of 88.8% indicates that the majority of claims in the answers are supported by the context of the document provided.

4. CONCLUSION

This study successfully implemented RAG on a LLM for the development of an information service chatbot at the USTI using a hybrid retrieval mechanism that combines FAISS and BM25 through Reciprocal Rank Fusion and cross-encoder reranking. The retrieval evaluation results indicate that the hybrid configuration achieves the best overall performance, with Precision@3 of 71.7%, Recall@3 of 87.5%, and NDCG@3 of 96.3%. Moreover, the result of the generation evaluation shows that the evaluation of the generation of the new generation of models proves that the combination of RAG leads to an improvement in the quality of the response over the LLM-only baseline, as indicated by the increase in BERTScore-F1 to about 89% and the faithfulness score of over 88%.

However, there are some limitations to this study, despite the encouraging results. First, the evaluation is carried out on a limited number of test queries, and the relevance annotation is carried out by a single annotator, which may result in some bias. Second, the evaluation is carried out on offline retrieval and generation quality, and does not include user-based evaluation, latency analysis, or system performance analysis. These limitations may be overcome by future studies that involve multiple annotators and larger evaluation datasets, and also include user studies and system performance analysis.

REFERENCES

[1] C. W. Okonkwo and A. Ade-Ibijola, "Chatbots applications in education: A systematic review," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100033, 2021, DOI: https://doi.org/10.1016/j.caeai.2021.100033.

[2] L. Huang *et al.*, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–55, Mar. 2025, DOI: https://doi.org/10.1145/3703155.

[3] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," Jul. 2020, DOI: https://arxiv.org/abs/2005.14165.

[4] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Apr. 2021, DOI: https://arxiv.org/abs/2005.11401.

[5] J. Swacha and M. Gracel, "Retrieval-Augmented Generation (RAG) Chatbots for Education: A Survey of Applications," *Applied Sciences*, vol. 15, no. 8, p. 4234, Apr. 2025, DOI: https://doi.org/10.3390/app15084234.

[6] M. Maryamah, M. M. Irfani, E. B. Tri Raharjo, N. A. Rahmi, M. Ghani, and I. K. Raharjana, "Chatbots in Academia: A Retrieval-Augmented Generation Approach for Improved Efficient Information Access," in *2024 16th International Conference on Knowledge and Smart Technology (KST)*, IEEE, Feb. 2024, pp. 259–264. DOI: https://doi.org/10.1109/KST61284.2024.10499652.

[7] A. Stolfo, "Groundedness in Retrieval-augmented Long-form Generation: An Empirical Study," in *Findings of the Association for Computational Linguistics: NAACL 2024*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 1537–1552. DOI: https://doi.org/10.18653/v1/2024.findings-naacl.100.

[8] L. R. Hidayat, I. G. P. S. Wijaya, and R. Dwiyansaputra, "Optimalisasi Layanan Sistem Informasi Mahasiswa dengan Integrasi Telegram : Chatbot Retrieval-Augmented-Generation Berbasis Large Language Model," *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTIKA)*, vol. 7, no. 1, pp. 121–131, Mar. 2025, DOI: https://doi.org/10.29303/jtika.v7i1.459.

[9] T. Muhammad, R. Rahardiansyah, R. Setya Perdana, and T. N. Fatyanosa, "Analisis Teknik Embedding Model NV-Embed pada Large Language Models Berbasis Retrieval Augmented Generation," 2025. [Online]. Available: http://j-ptiik.ub.ac.id.

[10] H. Tohir, N. Merlina, and M. Haris, "Utilizing Retrieval-Augmented Generation in Large Language Models to Enhance Indonesian Language NLP," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 10, no. 2, pp. 352–360, Nov. 2024, DOI: https://doi.org/10.33480/jitk.v10i2.5916.

[11] I. Pujiono, I. M. Agtyaputra, and Y. Ruldeviyani, "Implementing Retrieval-Augmented Generation and Vector Databases for Chatbots in Public Services Agencies Context," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 10, no. 1, pp. 216–223, Aug. 2024, DOI: https://doi.org/10.33480/jitk.v10i1.5572.

[12] R. D. Pesl, J. G. Mathew, M. Mecella, and M. Aiello, "Retrieval-Augmented Generation for Service Discovery: Chunking Strategies and Benchmarking," May 2025, DOI: https://arxiv.org/abs/2505.19310.

[13] S. F. Chaerul Haviana, M. Agus Riyadi, and R. Kusumaningrum, "Evaluation of Chunking Strategies in RAG Application for Explicit Retrieval on Indonesian Language Scientific Papers," in *2025 12th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, IEEE, Sep. 2025, pp. 59–65. DOI: https://doi.org/10.1109/EECSI67060.2025.11290624.

[14] K. Wei *et al.*, "PPN: Parallel Pointer-based Network for Key Information Extraction with Complex Layouts," Jul. 2023, DOI: https://doi.org/10.48550/arXiv.2307.10551.

[15] L. Wang *et al.*, "Text Embeddings by Weakly-Supervised Contrastive Pre-training," Feb. 2024, DOI: https://doi.org/10.48550/arXiv.2212.03533.

[16] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," Feb. 2020, DOI: https://arxiv.org/abs/1904.09675.

[17] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "Ragas: Automated Evaluation of Retrieval Augmented Generation," Apr. 2025. DOI: https://doi.org/10.48550/arXiv.2309.15217.