Retrieval-Augmented Generation Approach: Document Question Answering using Large Language Model

Kurnia Muludi[1], Kaira Milani Fitria[2]*, Joko Triloka[3], Sutedi[4]
Informatics Engineering Graduate Program, Darmajaya Informatics and Business Institute, Bandar Lampung, Indonesia

***Abstract*—This study introduces the Retrieval Augmented Generation (RAG) method to improve Question-Answering (QA) systems by addressing document processing in Natural Language Processing problems. It represents the latest breakthrough in applying RAG to document question and answer applications, overcoming previous QA system obstacles. RAG combines search techniques in vector store and text generation mechanism developed by Large Language Models, offering a time-efficient alternative to manual reading limitations. The research evaluates RAG's that use Generative Pre-trained Transformer 3.5 or GPT-3.5-turbo from the ChatGPT model and its impact on document data processing, comparing it with other applications. This research also provides datasets to test the capabilities of the QA document system. The proposed dataset and Stanford Question Answering Dataset (SQuAD) are used for performance testing. The study contributes theoretically by advancing methodologies and knowledge representation, supporting benchmarking in research communities. Results highlight RAG's superiority: achieving a precision of 0.74 in Recall-Oriented Understudy for Gisting Evaluation (ROUGE) testing, outperforming others at 0.5; obtaining an F1 score of 0.88 in BERTScore, surpassing other QA apps at 0.81; attaining a precision of 0.28 in Bilingual Evaluation Understudy (BLEU) testing, surpassing others with a precision of 0.09; and scoring 0.33 in Jaccard Similarity, outshining others at 0.04. These findings underscore RAG's efficiency and competitiveness, promising a positive impact on various industrial sectors through advanced Artificial Intelligence (AI) technology.**

Keywords*—*Natural Language Processing; Large Language Model; Retrieval Augmented Generation; Question Answering; GPT

I. Introduction

This research proposes a new approach to the increasing reliance on articles and journal documents by introducing a Question-Answering (QA) document processing system [1]. The identification of several critical problems motivates this research. The problems motivating this research are multifaceted. Firstly, manual reading and processing to comprehend document text are time-consuming, error-prone, and inefficient. Secondly, previous methods employed to modify Large Language Models (LLM) for document processing demanded substantial resources and were challenging to implement widely. Lastly, models relying solely on the capabilities of LLM for QA systems without modifications tend to generate hallucinatory answers, lacking correctness and precision. Manual processing for document understanding leads to time-consuming efforts, susceptibility to human error, and inefficient analysis processes. Based on previous methods, the use of modified Large Language Models (LLM) for document processing requires significant resources and poses challenges for widespread implementation. Also, the underutilization of the recently discovered Retrieval Augmented Generation (RAG) method, particularly in document processing within Question-Answering (QA) systems, provides an opportunity for further exploration. The motivation stems from the challenges associated with manual document processing, resource-intensive Large Language Model (LLM) modifications, and the underutilization of the Retrieval-Augmented Generation (RAG) method in the document-based question-answering domain [2], [3]. In addition, there is a tendency to produce hallucinatory responses that lack accuracy and precision in models that rely solely on LLM capabilities for QA systems without modifications. Finally, the implementation of RAG in QA systems for document processing offers the untapped potential to improve the ability of the system to produce accurate and non-hallucinatory responses.

Building on this line of research, this paper proposes the implementation of the Retrieval Augmented Generation model for document question answering tasks, specifically using the ChatGPT model. RAG, introduced in 2021 [4], addresses the limitations of previous methods by merging parametric and non-parametric memory. This hybrid model seamlessly integrates generative capabilities with data retrieval mechanisms, linking language models to external knowledge sources. RAG combines generative capabilities and the ability to search for data and incorporate relevant information from the knowledge base in the model. The distinct advantages of RAG lie in its ability to adapt to dynamic data, its flexibility in working with external data sources, and its ability to mitigate hallucinatory responses [5]. These characteristics make RAG particularly suitable for QA tasks on internal organizational documents by leveraging external knowledge to reduce response hallucinations [6].

The current research aims to exploit the innovative approach of RAG to construct an application capable of automatically processing external text documents. The focus of this research is to develop an application system capable of processing external document text uploaded by the user. The system will automatically read the document text, allowing users to input questions related to the document. Subsequently, the system provides answers based on the processed document

text, eliminating the need for manual reading to find answers. This comprehensive solution not only overcomes the limitations of previous methods, but also promises to significantly speed up research and study exploration in various domains.

Testing of the proposed model is performed, like several previous QA-based studies, by calculating the suitability of the answer results provided by the model with the ground truth of the test dataset. Some of the metrics used to calculate the performance of this model include Accuracy, ROUGE, BLEU, BERTScore, and Jaccard Similarity.

II. Related Works

This study examines the applicability of RAG, its impact on the document processing task, and compares it to the previous methods. This research also investigates the capability of the large language model within the ChatGPT systems, gpt-3.5-turbo within the framework of RAG. This work also highlights the development of Artificial Intelligence (AI) and Natural Language Processing (NLP), so this research focuses on the improvement of intelligence and the capabilities of applications [7], [8], [9]. Machine Learning and Deep Learning algorithms, which include BERT Base, and Text-to-Text Transfer Transformer (T5) models, and RAG method, have made significant advances in QA tasks [4], [10], [11], [12]. This research motivated the implementation of RAG for processing documents, integrated into an interactive QA system.

Between 2015 and the present, the evolution of question-answering (QA) systems shows a trajectory characterized by diverse methodologies. Starting with semantic parsing-based systems in 2015, Wen-tau Yih et al. focused on transforming natural language queries into structured logical forms, achieving a performance of 52.5% in the F1-score [2]. Subsequent knowledge-based paradigms (KB-QA) by Yanchao Hao et al. in 2017 reformulated questions as predicates, achieving a performance of 42.9% [3]. Progress has been made in integrating AI technologies. Caiming Xiong's exploration of dynamic memory networks (DMN) in 2016, achieved an accuracy of 28.79% [7]. In the same year, Minjoon Seo et al.'s Bi-Directional Attention Flow (BiDAF) framework demonstrated significant performance with a 68% exact match and 77.3% F1 score, albeit with a computational time of 20 hours [8]. Adams Wei Yu et al. introduced the QANet model in 2018, with a performance of 76.2% exact match and 84.6% F1-Score, within a shorter computational time of 3 hours [9]. As QA systems evolve, in 2019 Wei Yang et al. applied fine-tuning methods with data augmentation techniques, achieving remarkable results with a modified BERT-Base model of 49.2% for exact match and 65.4% for F1-Score [10]. Colin Raffel et al. introduced the Text-to-Text Transfer Transformer (T5), with impressive performance of 63.3% for exact match, 94.1% for F1 score, and a peak accuracy of 93.8%, albeit with an increased number of parameters of 11 billion [11]. In 2020, the focus was on fine-tuning pre-trained models, with Adam Roberts et al. achieving a recall performance of 34.6% using the T5 model [12]. The Retrieval-Augmented Generation (RAG) method, which combines parametric and non-parametric methods, was introduced by Patrick Lewis et al. in 2021. RAG has demonstrated its capabilities in open domain QA tasks, overcoming previous limitations to deliver more efficient and comprehensive QA systems [4].

Large language model called GPT, or Generative Pre-Trained Transformer was developed by OpenAI. Previous research that has compared the performance of ChatGPT with other large language models like PaLM and LLaMA in open-domain QA tasks indicates that ChatGPT consistently achieves the highest scores across various open-domain QA datasets [13]. Table I presents performance comparisons among LLMs.

TABLE I. LLM Performance on Open Domain QA Dataset

Model	TriviaQA	WebQuestion	NQ-Open
PaLM-540B (few-shot)	81.4	43.5	39.6
PaLM-540B (zero-shot)	76.9	10.6	21.2
LLaMA-65B (zero-shot)	68.2	-	23.8
ChatGPT (zero-shot)	85.9	50.5	48.1

The PolyQuery Synthesis test, which identifies multiple queries within a single-query prompt and extracts the answers to all of the questions from the model's latent representation, also shows that ChatGPT outperforms other GPT models from OpenAI (ada-001, babbage-001, curie, and davinci) in terms of accuracy [13]. According to the evaluations, the gpt-3.5-turbo model has been selected for implementation in this research.

III. Research Method

This research undergoes a development phase, starting with designing the application system and integrating the APIs of ChatGPT, LangChain and FAISS. Subsequent stages include extensive system modeling, interface testing and data preparation using the proposed dataset and the SQuAD dataset. The testing phase, which includes a performance comparison with other applications using ground truth metrics (ROUGE, BERTScore, BLEU and Jaccard Similarity), guides the exploration of the capabilities of the proposed system, as shown in Fig. 1.

A. RAG Integration

Retrieval Augmented Generation (RAG) combines retrieval and generation models. It uses a Large Language Model (LLM) to generate text based on commands and integrates information from a separate retrieval system to improve output quality and contextual relevance [14]. The mechanism involves retrieving factual content from a knowledge base via retrieval models and using generative processes to provide additional context for more accurate output [15]. External data sources are used, and the numerical representation is facilitated by embedding methods to ensure compatibility. Based on Fig. 2, user queries converted into embeddings are compared with vectors from the knowledge library. Relevant context is added to the queries before they are fed into the base language model.

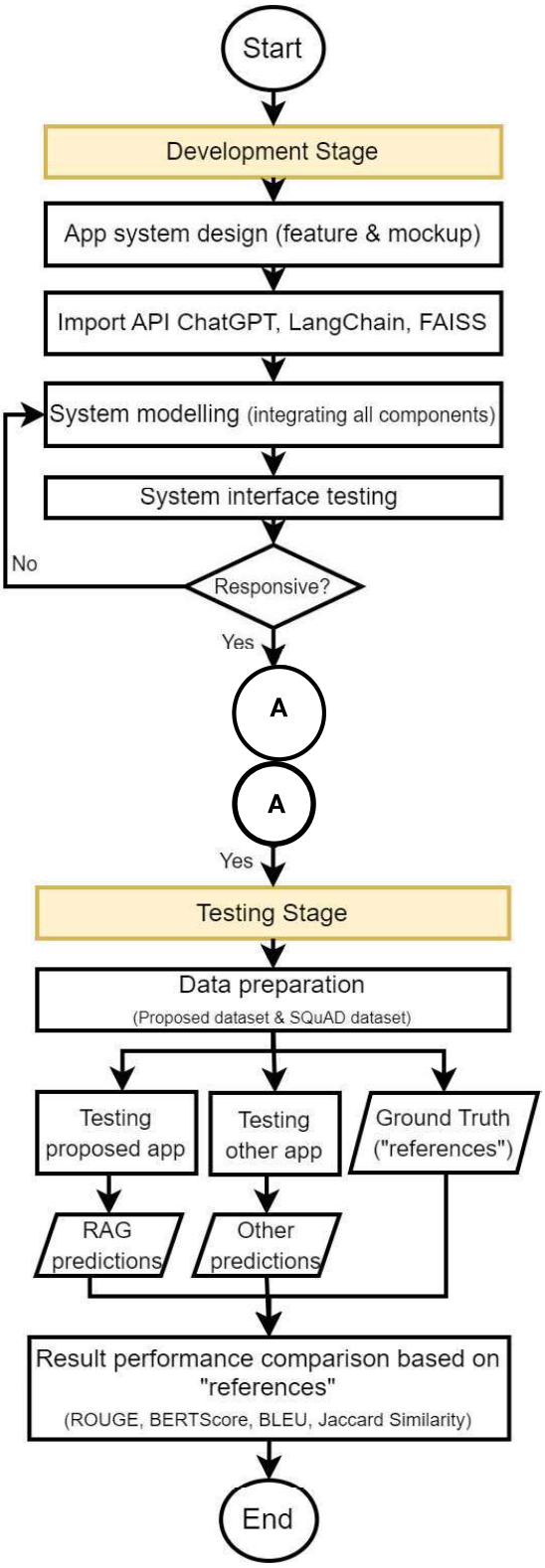

Fig. 1. Research flow diagram.

OpenAI, the creator of the Large Language Model GPT, conducted a comprehensive number of RAG experiments, exploring various implementations such as cosine similarity retrieval, chunk/embedding experiments, reranking, classification steps, and prompt engineering, as depicted in Fig. 3. OpenAI's findings, presented in Fig. 3, revealed that RAG implementation with prompt engineering achieved the highest accuracy, positioning it as the most effective RAG technique to date [16]. This discovery serves as a catalyst for the integration of RAG with prompt engineering using the LangChain module.

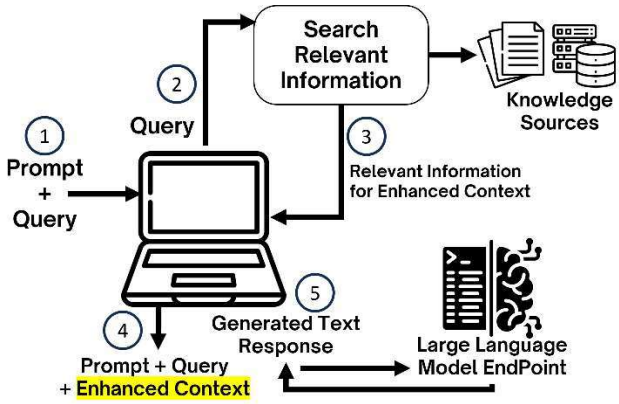

Fig. 2. RAG mechanism with LLM.

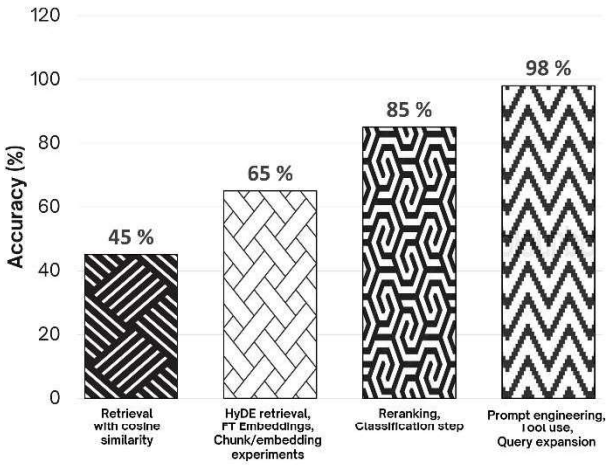

Fig. 3. Accuracy of the RAG method by Open AI.

LangChain provides a robust data processing pipeline that utilizes FAISS to perform an efficient retrieval operation in the VectorDB. The query phase transforms inputs into vectors for database searches, and prompt engineering enhances the reusability of retrievals. Output parsers interpret LLM outputs, ensuring consistency [17]. A highly efficient similarity search and vector clustering library, Facebook AI Similarity Search or FAISS [18]. It optimizes the trade-off between memory, speed and accuracy, allowing developers to effectively navigate multimedia documents. The mechanism involves the construction of an index for efficient storage, with vector searches retrieving the most similar vectors using cosine similarity scores [19].

B. Proposed Model

This research employs a modified Large Language Model (LLM), ChatGPT, augmented with additional libraries to function as a Question-Answering (QA) system capable of processing external documents for supplementary information.

The chosen methodology for QA system development is the Retrieval Augmented Generation (RAG) mechanism. Unlike previous approaches such as semantic parsing-based, knowledge-based, and fine-tuning using LSTM or other DL algorithms, RAG addresses shortcomings like difficulty expanding or revising model memory, an inability to provide direct insight into generated predictions, and a tendency to produce hallucinative answers [12]. The solution involves the creation of a hybrid model, merging generative and retrieval models, which forms the basis for the RAG method. RAG offers advantages such as adaptive responses to dynamic data, flexibility with external data sources, and minimization of hallucinative responses [5]. Thus, RAG is chosen to construct a text document-based QA system interacting with users through a chatbot interface. The system's workflow, implemented using RAG and supporting libraries like LangChain and FAISS, is illustrated in Fig. 4.

The integration of the LangChain framework into the QA document system includes document loading, memory management, and prompting to connect to the LLM model. The process starts with document loading, followed by document splitting into text chunks. These text chunks undergo word embedding, converting them into vectors stored in the vector database. Simultaneously, user-inputted text questions are embedded and converted into word vectors. The system connects these vectors to the vector database, performing a semantic search and ranking the relevance between vectors. The semantic search results in relevant context between questions and answers. The system retrieves pertinent answers based on user queries and sends them to the LLM (using the ChatGPT model). The final outcome involves the system receiving LLM-generated answers and delivering them to the user. The application system interacts with users, requiring an interface connecting the user and the system. Mockups, design layouts, and elements for the web application are created using the Streamlit framework, facilitating rapid development and sharing of the AI model web application. The mockup for the application system and user interaction within the system is depicted in Fig. 5.

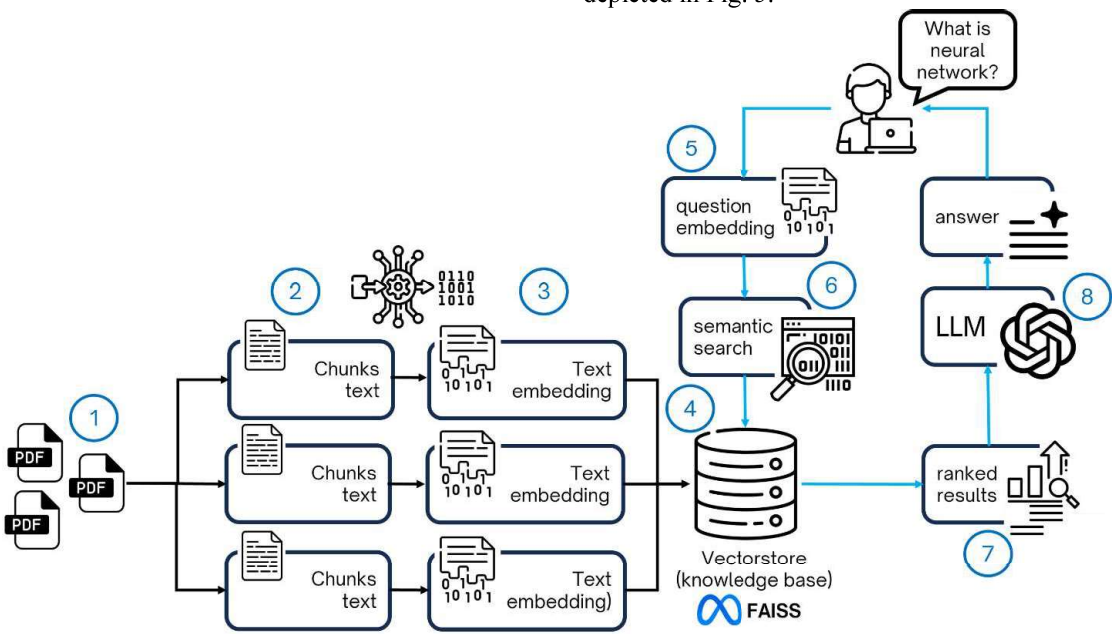

Fig. 4. Integration of langchain framework in RAG for the proposed document QA system.

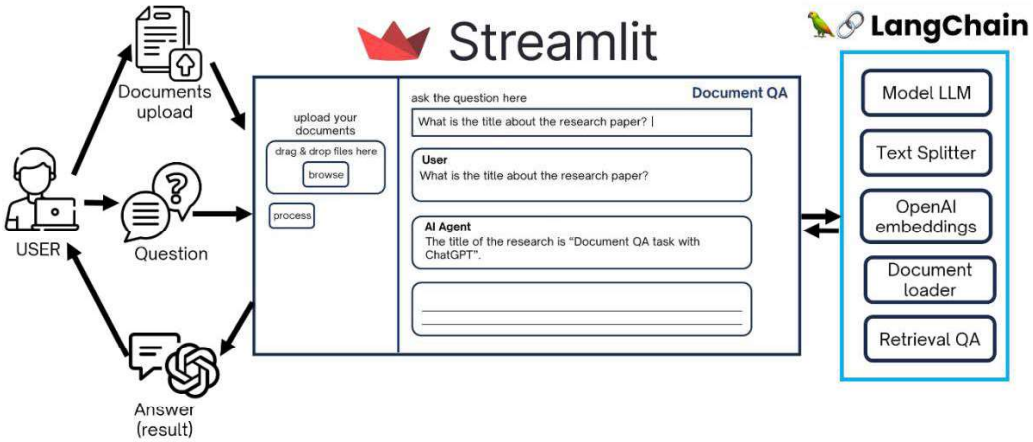

Fig. 5. Mockup of the application system and user interaction for the app.

C. Proposed Dataset DocuQA

The proposed dataset, DocuQA, designed for application-based question-answering systems that process document inputs, consists of 20 diverse documents, encompassing journal articles, news reports, financial documents, and tutorials. Each document file includes five questions with corresponding ground truth answers, enabling a thorough evaluation of QA system capabilities, with a total 100 questions in the dataset. DocuQA consists of journal documents with calculations and formulas, news documents with specific titles, financial reports and news documents with numbers and currency data, and tutorial documents with step-by-step instructions. Accuracy can be calculated based on the correct answers out of 100, providing a metric for information extraction accuracy. The dataset aims to challenge QA systems in understanding context, identifying keywords, and efficiently extracting specific information, offering a robust evaluation tool for developers and researchers across various document and question types. The dataset can be accessed publicly [20].

Proper citation of the dataset is encouraged for research or projects using DocuQA to ensure appropriate credit is given. The preview of the DocuQA dataset can be seen in Fig. 6.

Files	Question	Ground Truth
R4	Can you inform the key numbers of fourth-quarter vehicle production and deliveries report for 2023 from Tesla?	Total deliveries Q4 2023 is 484.507 Total production Q4 2023 is 494.989 Total annual deliveries 2023 is 1.808.581 Total annual production 2023 is 1.845.985
R4	How many electric vehicle deliveries and production based on Tesla's report in 2022?	1.31 million deliveries and 1.37 million production
R4	How many units of Chinese automaker BYD's new energy vehicles were sold in 2023?	3.02 million
R5	what is the title of the news report?	Copper could skyrocket over 75% to record highs by 2025 — brace for deficits analysts say
R5	when the news published?	January 2 2024

Fig. 6. Preview of the DocuQA test dataset.

D. Testing and Evaluation

The tests were performed on two types of test datasets, with DocuQA [20] and SQuAD 1.1 [21]. DocuQA is a dataset originally created by this research, consisting of 100 questions with the ground truth and a total of 20 test documents for document-based QA systems. In addition, the SQuAD dataset was used in the form of modified pdf documents that can be used to test the QA system's ability to process documents and retrieve information based on the questions and related ground truth in the SQuAD dataset. Both types of test datasets will be tested on the QA system developed in this research, and also on other commercial QA systems that process pdf documents, such as typeset.io. The results of these tests will give an idea of the QA system performance built on this research, whether it is superior to other document-based QA applications.

The proposed QA document processing system is evaluated through rigorous testing using established metrics such as ROUGE or Recall-Oriented Understudy for Gisting Evaluation, BERTscore, BLEU or Bilingual Evaluation Understudy, and Jaccard Similarity. These metrics provide reliable benchmarks for assessing the system's performance across various dimensions. The testing process involves two key variables. "Predictions RAG" and "Prediction Others" represent the test results from the developed application and comparable commercial applications, respectively. Both sets of predictions are compared to the ground truth data, which is encapsulated in the "references" variable. Different aspects of language models and question answering systems are evaluated using different metrics. ROUGE measures the overlap in summarization [22]. BERTscore assesses semantic similarity using contextual embeddings [23]. BLEU evaluates n-gram precision [24], and Jaccard Similarity compares text similarity based on word or n-gram overlap [25]. Precision in question answering systems is commonly assessed through accuracy, F1 score, and precision metrics, providing insights into their effectiveness. The metrics are used to quantitatively evaluate system performance and establish its superiority over existing commercial applications in document processing and information retrieval tasks.

1) Accuracy: Accuracy is defined as the proportion of correct responses from the total number of responses. Accuracy can be calculated by calculating the percentage of correct predictions over the total number of references [26]. In essence, accuracy represents the ability of the system to provide correct answers, which is expressed as a percentage using the following formula (see Eq.(1)).

$$\text{Accuracy} = \frac{\text{correct predictions}}{\text{all predictions}} \times 100\% \quad (1)$$

This metric serves as a valuable indicator of the overall correctness of the model in the response it generates.

2) ROUGE: Recall-Oriented Understudy for Gisting Evaluation can be used to evaluate the text generation models, which are based on the measurement of the overlap between candidate text and reference text [27]. ROUGE has several measurement variants, each depending on the number of overlapping n-grams. The ROUGE-L variant is the most widely used, because it uses the longest sequence or longest common subsequence or LCS with the longest word sequence that both sentences have. Precision refers to the proportion of n-grams in the candidate that are also in the reference (see Eq. 2.). Recall, on the other hand, refers to the proportion of n-grams that are in the reference text that exactly match in the predicted candidate text as shown in Eq. (3). The F1-score can be calculated from the precision and recall as shown in Eq. (4).

$$\text{ROUGE-L}_{\text{recall}} = \frac{\text{LCS (candidate, reference)}}{\#\text{words in reference}} \quad (2)$$

$$\text{ROUGE-L}_{\text{precision}} = \frac{\text{LCS (candidate, reference)}}{\#\text{words in candidate}} \quad (3)$$

$$\text{ROUGE-L}_{\text{F1-Score}} = 2 \times \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \quad (4)$$

where, the reference is based on the ground truth in the test dataset, and the candidate is from the system predictions. The score generated by the ROUGE measure is between 0 and 1. A

score of 1 indicates total agreement between reference and candidate text.

3) BERTScore: BERTScore is an automatic evaluation metric in text generation tasks that evaluates the similarity of each candidate sentence token to each reference sentence token by means of contextual embeddings [23]. The embeddings in BERTScore are contextual, changing depending on the sentence context. The context awareness allows BERTScore to score semantically similar sentences despite their different sentence order. For the recall calculation, each token in x is matched with the most similar token in $\hat{x}$, as for the precision calculation. Greedy matching is used to maximize the similarity score. The values of precision (see Eq. (5)), recall (see Eq. (6)) and F1 score (see Eq. (7)) for reference x and candidate $\hat{x}$ can be calculated using the following equations.

$$R_{BERT} = \frac{1}{|x|}\sum_{x_i \in x} \max_{\hat{x}j \in \hat{x}} x_i^T \hat{x}_j \quad (5)$$

where, R_{BERT} is the Recall BERTScore, x is the reference token, $\hat{x}$ is the candidate token, x_i is the sequence vector x, x_j is the sequence vector $\hat{x}$, where $\Sigma_{x_i \in x}$ is the number of x_i present in x, and also $\max_{\hat{x}_j \in \hat{x}}$ is the maximum value of $\hat{x}_j$ present in $\hat{x}$, and $x_i^T \hat{x}_j$ is the cosine similarity of x and $\hat{x}$.

$$P_{BERT} = \frac{1}{|\hat{x}|}\sum_{\hat{x}_j \in \hat{x}} \max_{xi \in x} x_i^T \hat{x}_j \quad (6)$$

Given P_{BERT} as Precision BERTScore, x as reference token, $\hat{x}$ as candidate token, x_i as sequence vector x, $\hat{x}_j$ as sequence vector $\hat{x}$, where $\Sigma_{\hat{x}_j \in \hat{x}}$ is the number of $\hat{x}_j$ present in $\hat{x}$, and also $\max_{x_i \in x}$ is the maximum value of x_i present in x, and $x_i^T \hat{x}_j$ is the cosine similarity of x and $\hat{x}$.

$$F_{BERT} = 2 \times \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \quad (7)$$

where F_{BERT} is the F1-score of BERTScore, then P_{BERT} is the precision and R_{BERT} is the recall from BERTScore results. Although the cosine similarity value is theoretically in the interval [-1, 1], in practice the value is rescaled so that it is between 0 and 1 in the result of the BERTScore calculation.

4) BLEU: Bilingual Evaluation Understudy is a metric that computes a modification of precision for n-grams, combines it with weights, and applies a brevity penalty to obtain the final BLEU score [28]. The score range of BLEU is from 0 to 1. The greater the BLEU score, the better the system's performance is considered to be compared to the references. The formula for calculating BLEU can be seen in Eq. (8).

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (8)$$

BP represents the brevity penalty, adjusting the score to penalize translations shorter than the reference. *N* denotes the maximum number of considered n-grams. The precision for n-grams, denoted as p_n signifies the n-grams ratios by the candidate text that appearing in any reference translation to the total of n-grams in the candidate text. w_n represents the weight assigned to each n-gram precision score.

5) The Jaccard similarity quantifies the similarity percentage between two sets of data by identifying the common and the different members [29]. This can be calculated by dividing the number of observations shared by the sum of the observations in each of the two sets. Jaccard similarity can be expressed as the ratio of the intersection $(A \cap B)$ to the union $(A \cup B)$ of two sets (see Eq. (9)).

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (9)$$

$|A \cap B|$ indicates the size of the intersection of the sets A and B, and $|A \cup B|$ indicates the size of the union of the sets A and B. The Jaccard similarity is bounded in the range from 0 to 1. A Jaccard similarity of 1 indicates complete identity between the sets, while a similarity of 0 implies that the sets have no common elements.

IV. Result and Discussion

A. Result

The interface of the proposed QA system can be seen in Fig. 7.

Fig. 7. Document QA system interface.

The interface of the proposed document QA system can accept multiple PDF format documents. If the user clicks the submit button, the system will process the PDF document to convert it to vector form with embedding (as described in the RAG mechanism in Fig. 2). Once the document submission process is complete, the user can ask questions related to the submitted document, and the QA system will provide answers based on the source documents provided. The set of questions and answers generated from the user's interaction with the QA system will be in the form of a chatbot, so that it stores the communication history.

B. Accuracy

Accuracy in our system model is expressed as the percentage of correct answers within the entire answer key dataset. To assess accuracy, we calculate the ratio of the number of correct predictions to the total number of predictions [26]. The visualization of this accuracy result can be figured in Fig. 8.

The accuracy comparison between the proposed QA document system and other applications reveals the superiority of our method. The proposed system achieved accuracy rates of 96% (our dataset) and 95.5% (SQuAD dev dataset), surpassing the other application's rates of 55% (our dataset) and 85.7% (SQuAD dataset). This underscores the consistently higher accuracy of our proposed method.

C. ROUGE

ROUGE-L score evaluation compares the results of our proposed QA method outperforming other QA applications in terms of precision, recall, and F1-Score. Specifically, on our dataset, our proposed method demonstrated precision, recall, and F1-Score of 73.7%, 23.9%, and 33.7%, respectively. In comparison, other QA applications achieved lower performance metrics with precision, recall, and F1-Score of 50.0%, 10.5%, and 15.2%, respectively. Similarly, on the SQuAD dev dataset, our proposed method excelled with precision, recall, and F1-Score reaching 85.5%, 16.2%, and 26.1%, while other QA applications reported lower scores of 77.2%, 10.4%, and 17.1%, respectively. These results underscore the superior performance of our proposed method across both datasets that can be visualized in Fig. 9.

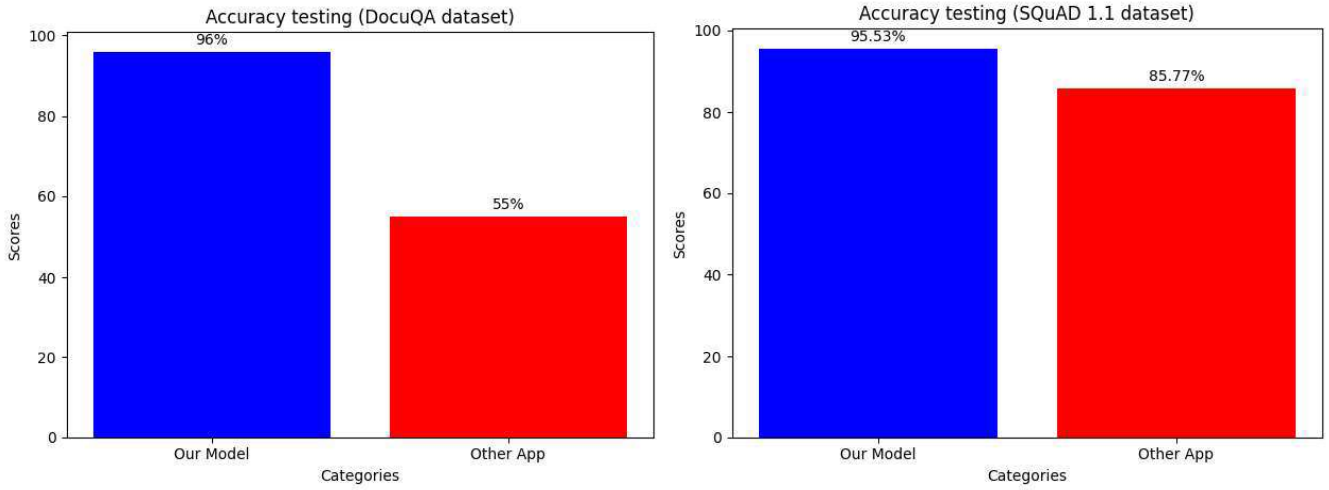

Fig. 8. Accuracy result of proposed method using RAG and other document QA application.

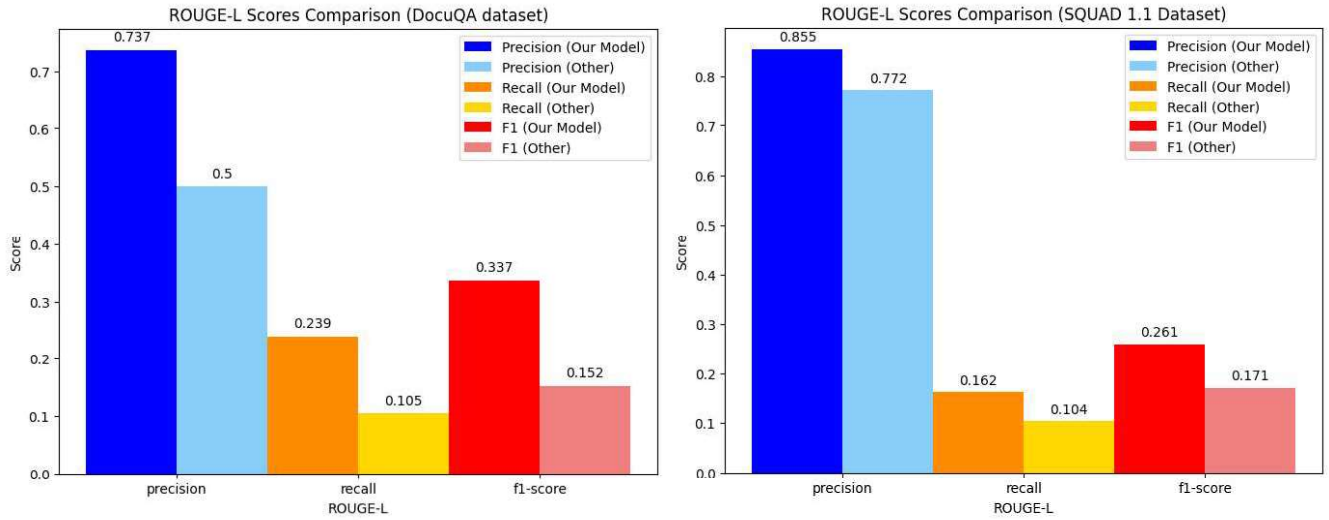

Fig. 9. ROUGE-L result of proposed method using RAG and other document QA application.

D. BERTScore

BERTScore evaluation compares the results of our proposed QA method outperforming other QA applications in terms of precision, recall, and F1-Score. Specifically, on our dataset, our proposed method demonstrated precision, recall, and F1-Score of 85.2%, 90.1%, and 87.6%, respectively. In comparison, other QA applications achieved lower performance metrics with precision, recall, and F1-Score of 81.6%, 86.3%, and 83.8%, respectively. Similarly, on the SQuAD dev dataset, our proposed method excelled with precision, recall, and F1-Score reaching 82.8%, 87.0%, and 84.8%, while other QA applications reported lower scores of 80.4%, 86.3%, and 83.2%, respectively. These results underscore the superior performance of our proposed method across both datasets that can be visualized in Fig. 10.

E. BLEU Accuracy

The BLEU metric score taken is the precision value, to capture the ability of each model to extract keyword answers that match the ground truth. Specifically, on our dataset, our proposed method demonstrated precision of 28.2%. In comparison, other QA applications achieved lower performance precision 9.7%. Similarly, on the SQuAD dev dataset, our proposed method excelled with precision 17.7%, while other QA applications reported lower scores of precision 5.6%. These results underscore the superior performance of our proposed method across both datasets that can be visualized in Fig. 11.

F. Jaccard Similarity

The performance of our QA system, as evaluated through Jaccard Similarity, is outstanding. Our method achieved 33.3% on our dataset and 11.1% on SQuAD dev using RAG method. In comparison, other QA applications scored lower with 4.1% on our dataset and 9.1% on SQuAD dev. These results highlight our method's superiority in Jaccard Similarity on both datasets that can be visualized in Fig. 12.

G. Discussion

The accuracy result of 95.5% in the SQuAD dev dataset outperforms other research with 61.5% accuracy that tested in SQuAD dev dataset [30] and 71.4% accuracy which also tested in SQuAD dev dataset [31]. We also using SQuAD dev dataset for testing the other document QA application platform, and it shows accuracy 85.7%. So, the model proposed in this study has a higher accuracy score compared to other applications, and previous research on the SQuAD test dataset.

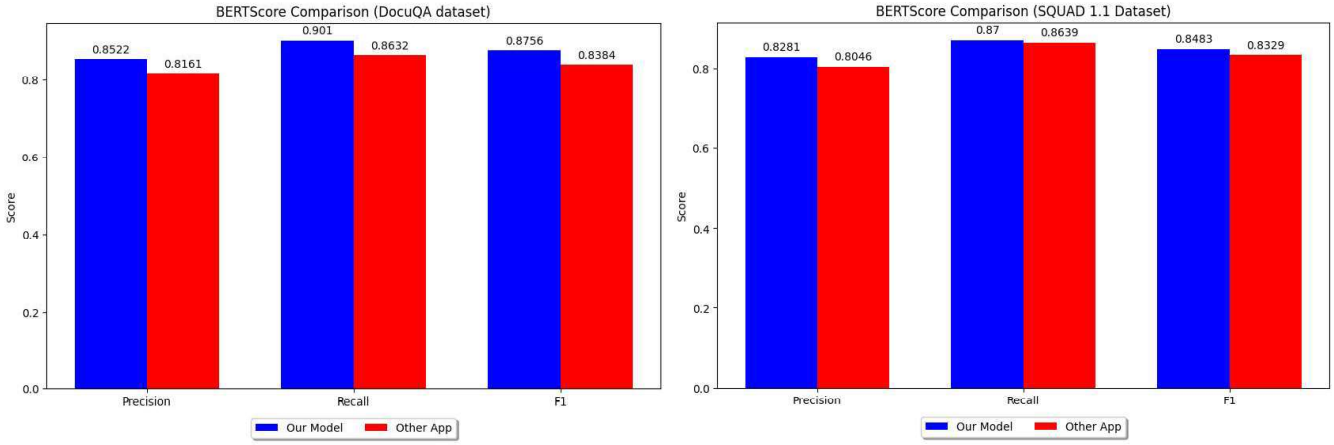

Fig. 10. BERTScore result of proposed method using RAG and other document QA application.

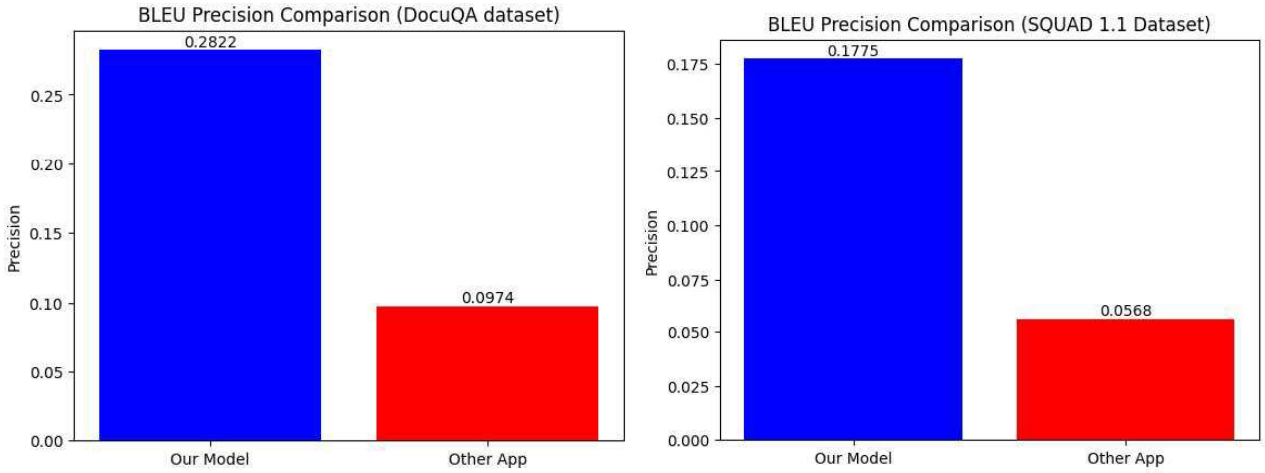

Fig. 11. BLEU precision result of proposed method using RAG and other document QA application.

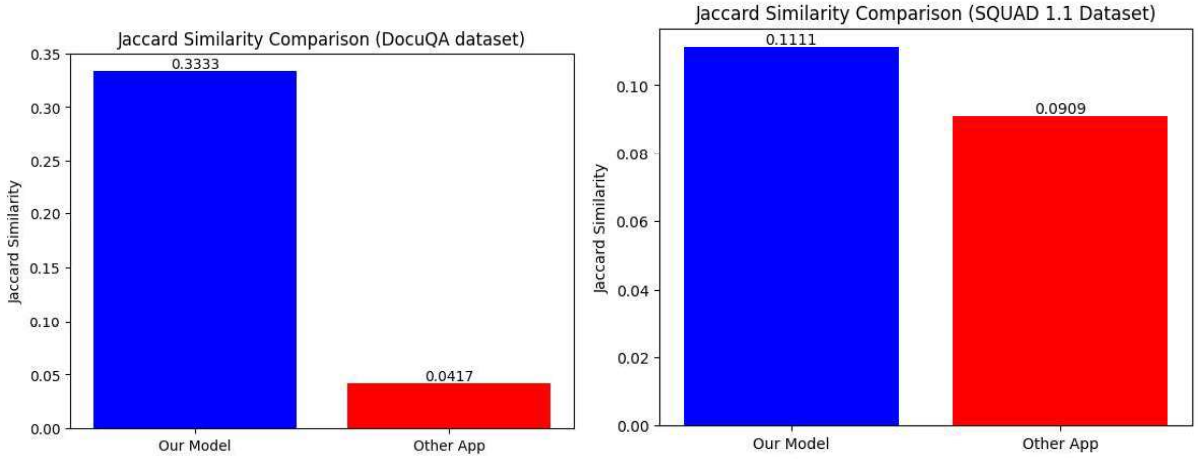

Fig. 12. Jaccard Similarity result of proposed method using RAG and other document QA application.

Our system's precision, recall, and F1-Score are 82.8%, 87%, and 84.8%, respectively, which surpass the precision of 62%, recall of 87%, and F1-Score of 67% reported in other research [32]. The proposed QA system's effectiveness is affirmed by the fact that it surpasses the recall result of other research with 42.70% [33] and outperforms other research [31], [34], [35] in terms of F1-Score, which is 42.6% [31], 49% [34], and 70.8% [35]. This positions it as a leading solution for automatic document processing and information retrieval tasks across a wide range of domains.

Based on the results of testing the proposed model, the results of the present study agree with previous literature studies, namely that the RAG method, through the implementation of a hybrid model combining parametric and nonparametric models, is able to provide good results [4]. In this case we combine the LangChain and FAISS frameworks for the RAG technique, and it can provide a good result. This model also combined with the use of the best language model at this current time like GPT-3.5, which provides good results. This is a very interesting performance that should be further developed.

V. Conclusion

Our proposed model for Question-Answering (QA) document processing integrates the Retrieval-Augmented Generation (RAG) model. The evaluation of our proposed QA system demonstrates its superiority over existing commercial applications in terms of Accuracy, ROUGE-L scores, BERTScore metrics, BLEU precision, and Jaccard Similarity. The proposed method achieved high accuracy rates of 96% and 95.5% on our dataset and the SQuAD dev dataset, respectively, outperforming other applications tested on the same datasets. Our system's precision, recall, and F1-Score metrics were superior to those of other QA applications on both datasets, as highlighted by the ROUGE-L evaluation. Additionally, the BERTScore metrics consistently showed higher precision, recall, and F1-Score for our proposed method compared to other applications. In addition, our QA system has demonstrated superior performance in keyword extraction and text similarity compared to other applications, as assessed by BLEU precision and Jaccard Similarity.

VI. Future Works

In the future, studies could be conducted to refine the architecture of the system, explore additional ways of using external data, and improve the scalability of the model for broader applications. The integration of user feedback mechanisms and continuous learning modules could contribute to the adaptability of the system and further improve its accuracy over time. In addition, exploring ways of processing documents in real time and extending the system's compatibility with different document formats could open up new opportunities for research and study.

References

[1] F. Ganier and R. Querrec, "TIP-EXE: A Software Tool for Studying the Use and Understanding of Procedural Documents," IEEE Trans Prof Commun, vol. 55, no. 2, pp. 106–121, Jun. 2012, doi: 10.1109/TPC.2012.2194600.

[2] W. Yih, M.-W. Chang, X. He, and J. Gao, "Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Stroudsburg, PA, USA: Association for Computational Linguistics, 2015, pp. 1321–1331. doi: doi.org/10.3115/v1/P15-1128.

[3] Y. Hao et al., "An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 221–231. doi: 10.18653/v1/P17-1021.

[4] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems, vol. abs/2005.11401, pp. 9459–9474, May 2020, doi: 10.48550/arXiv.2005.11401.

[5] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, and S. Nanayakkara, "Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering," Trans Assoc Comput Linguist, vol. 11, pp. 1–17, 2023, doi: 10.1162/tacl_a_00530.

[6] Y. Ahn, S.-G. Lee, J. Shim, and J. Park, "Retrieval-Augmented Response Generation for Knowledge-Grounded Conversation in the Wild," IEEE Access, vol. 10, pp. 131374–131385, 2022, doi: 10.1109/ACCESS.2022.3228964.

[7] Xiong, S. Merity, and R. Socher, "Dynamic Memory Networks for Visual and Textual Question Answering," Proceedings of The 33rd

International Conference on Machine Learning, pp. 2397–2406, Mar. 2016, doi: 10.48550/arXiv.1603.01417.

[8] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional Attention Flow for Machine Comprehension," International Conference on Learning Representations, Nov. 2016, doi: 10.48550/arXiv.1611.01603.

[9] A. W. Yu et al., "QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension," International Conference on Learning Representations, vol. abs/1804.09541, Apr. 2018, doi: 10.48550/arXiv.1804.09541.

[10] W. Yang, Y. Xie, L. Tan, K. Xiong, M. Li, and J. Lin, "Data Augmentation for BERT Fine-Tuning in Open-Domain Question Answering," ArXiv, vol. abs/1904.06652, Apr. 2019, doi: 10.48550/arXiv.1904.06652.

[11] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research, vol. 21, pp. 140:1-140:67, 2019, doi: 10.48550/arXiv.1910.10683.

[12] A. Roberts, C. Raffel, and N. Shazeer, "How Much Knowledge Can You Pack Into the Parameters of a Language Model?," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 5418–5426. doi: 10.18653/v1/2020.emnlp-main.437.

[13] M. T. R. Laskar, M. S. Bari, M. Rahman, M. A. H. Bhuiyan, S. R. Joty, and J. Huang, "A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets," in Annual Meeting of the Association for Computational Linguistics, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258967462

[14] W. Yu, "Retrieval-augmented Generation across Heterogeneous Knowledge," in Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, Seattle, Washington: Association for Computational Linguistics, Jul. 2022, pp. 52–58. doi: 10.18653/v1/2022.naacl-srw.7.

[15] D. Thulke, N. Daheim, C. Dugast, and H. Ney, "Efficient Retrieval Augmented Generation from Unstructured Knowledge for Task-Oriented Dialog," Conference of Association for the Advancement of Artificial Intelligence (AAAI), Feb. 2021, doi: 10.48550/arXiv.2102.04643.

[16] OpenAI, "A Survey of Techniques for Maximizing LLM Performance." Nov. 2023.

[17] Jacob Lee, "Building LLM-Powered Web Apps with Client-Side Technology." Accessed: Dec. 01, 2023. [Online]. Available: https://ollama.ai/blog/building-llm-powered-web-apps

[18] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with GPUs," IEEE Trans Big Data, vol. 7, no. 3, pp. 535–547, 2021, doi: 10.1109/TBDATA.2019.2921572.

[19] J. Zhu, J. Jang-Jaccard, I. Welch, H. Al-Sahaf, and S. Camtepe, A Ransomware Triage Approach using a Task Memory based on Meta-Transfer Learning Framework. 2022. doi: 10.48550/arXiv.2207.10242.

[20] K. M. Fitria, "DocuQA: Document Question Answering Dataset." Feb. 2024. doi: 10.6084/m9.figshare.25223990.v1.

[21] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in Conference on Empirical Methods in Natural Language Processing, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:11816014

[22] A. Chen, G. Stanovsky, S. Singh, and M. Gardner, "Evaluating Question Answering Evaluation," in Proceedings of the 2nd Workshop on Machine Reading for Question Answering, A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, and D. Chen, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 119–124. doi: 10.18653/v1/D19-5817.

[23] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," International Conference on Learning Representations, vol. abs/1904.09675, Apr. 2019, doi: 10.48550/arXiv.1904.09675.

[24] B. Ojokoh and E. Adebisi, "A Review of Question Answering Systems," Journal of Web Engineering, vol. 17, no. 8, pp. 717–758, 2019, doi: 10.13052/jwe1540-9589.1785.

[25] J. Soni, N. Prabakar, and H. Upadhyay, "Behavioral Analysis of System Call Sequences Using LSTM Seq-Seq, Cosine Similarity and Jaccard Similarity for Real-Time Anomaly Detection," in 2019 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, Dec. 2019, pp. 214–219. doi: 10.1109/CSCI49370.2019.00043.

[26] J. F. BELL and A. H. FIELDING, "A review of methods for the assessment of prediction errors in conservation presence/absence models," Environ Conserv, vol. 24, no. 1, pp. 38–49, 1997, doi: DOI: 10.1017/S0376892997000088.

[27] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," Association for Computational Linguistics, vol. Text Summa, no. 12, pp. 74–81, 2004, [Online]. Available: https://aclanthology.org/W04-1013/

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, P. Isabelle, E. Charniak, and D. Lin, Eds., Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. doi: 10.3115/1073083.1073135.

[29] N. C. Chung, B. Miasojedow, M. Startek, and A. Gambin, "Jaccard/Tanimoto similarity test and estimation methods for biological presence-absence data," BMC Bioinformatics, vol. 20, no. 15, p. 644, 2019, doi: 10.1186/s12859-019-3118-5.

[30] A. Stricker, "Question answering in Natural Language: the Special Case of Temporal Expressions," in Proceedings of the Student Research Workshop Associated with RANLP 2021, S. Djabri, D. Gimadi, T. Mihaylova, and I. Nikolova-Koleva, Eds., Online: INCOMA Ltd., Sep. 2021, pp. 184–192. [Online]. Available: https://aclanthology.org/2021.ranlp-srw.26

[31] S. Min, V. Zhong, R. Socher, and C. Xiong, "Efficient and Robust Question Answering from Minimal Context over Documents," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), I. Gurevych and Y. Miyao, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1725–1735. doi: 10.18653/v1/P18-1160.

[32] H. Bahak, F. Taheri, Z. Zojaji, and A. Kazemi, "Evaluating ChatGPT as a Question Answering System: A Comprehensive Analysis and Comparison with Existing Models," ArXiv, vol. abs/2312.07592, Dec. 2023, doi: 10.48550/arXiv.2312.07592.

[33] T. Cakaloglu, C. Szegedy, and X. Xu, "Text Embeddings for Retrieval From a Large Knowledge Base," Research Challenges in Information Science, vol. abs/1810.10176, Oct. 2018, doi: 10.48550/arXiv.1810.10176.

[34] S. Gholami and M. Noori, "Zero-Shot Open-Book Question Answering," ArXiv, vol. abs/2111.11520, Nov. 2021, doi: doi.org/10.48550/arXiv.2111.11520.

[35] G. Nur Ahmad and A. Romadhony, "End-to-End Question Answering System for Indonesian Documents Using TF-IDF and IndoBERT," in 2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA), 2023, pp. 1–6. doi: 10.1109/ICAICTA59291.2023.10390111.