JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 2435-2445

A LOW CODE APPROACH TO Q&A ON CARE RECORDS USING FLOWISE AI WITH LLM INTEGRATION AND RAG METHOD

Defry Hamdhana*[1)]
1. Department of Informatics, Engineering, Universitas Malikussaleh, Indonesia

Article Info
Keywords: Care Records; Low-Code Approach; Flowise AI; RAG; Information Retrieval

Article history:
Received 10 November 2024
Revised 19 November 2024
Accepted 30 November 2024
Available online 2 December 2024

DOI :
https://doi.org/10.29100/jipi.v9i4.6978

* Corresponding author.
Defry Hamdhana
E-mail address:
defryhamdhana@unimal.ac.id

ABSTRACT
Care records are vital for monitoring patient conditions and supporting clinical decision-making, but their diverse formats—such as tables, narrative sentences, checklists, and fill-in-the-blank fields—present challenges for efficient information retrieval. Traditional retrieval methods are often time-consuming and error-prone, while automated systems struggle with contextual accuracy in complex medical language. This study proposes a low-code approach to develop a question-and-answer (Q&A) system for care records using Flowise AI integrated with Retrieval-Augmented Generation (RAG) methodology. By utilizing LangChain and OpenAI's language models, Flowise AI provides a framework for constructing a Q&A system that retrieves information accurately across different documentation formats. The system employs components such as Recursive Character Text Splitter, PDF processing, OpenAI Embeddings, In-Memory Vector Store, and a Conversational Retrieval QA Chain, ensuring efficient retrieval with contextual relevance. Our results demonstrate high accuracy in aligning the Q&A responses with ground truth data, validating the system's effectiveness in healthcare documentation retrieval. This low-code solution not only enhances accessibility for non-technical users but also empowers healthcare professionals with a scalable tool for quick access to critical patient data. The findings underscore the potential of low-code AI systems like Flowise AI, utilizing RAG, to improve information retrieval in healthcare, supporting more accurate and timely clinical decisions.

I. INTRODUCTION

Care records play a vital role in ensuring the quality of patient care. Through these records, healthcare workers can record, monitor, and evaluate patient conditions in detail[1]. The recorded information, such as changes in vital signs, treatment history, treatment actions, and responses to medical interventions, form the basis for making appropriate clinical decisions. Therefore, effective utilization of these records is essential to improve the quality of healthcare services and ensure patient safety[2]. Furthermore, care records also useful to predict patient health in the future so that doctors can immediately anticipate things that might become problems for patients in the future[3]. However, one of the main challenges in managing care records is the complexity such as the various formats of care records and the information that needs to be obtained quickly[4]. Traditionally, information retrieval from records has been done manually or semi-automatically, which tends to be time-consuming and prone to human error. In addition, the lack of an efficient automated system prevents healthcare workers from accessing important information quickly, which ultimately impacts decision making in handling patient conditions. Semi-automated systems, while better, often lack contextual awareness and struggle to manage complex medical language. Several studies have explored the use of Natural Language Processing (NLP) to address these challenges, such as Lamy et al who used NLP and information extraction (IE) to automate the extraction of clinical information, reducing manual workload and potential human error[5]. Due to the complexity and specificity of medical language and data, this research requires further development. In another study, Rybinski et al also used NLP by using domain adaptive pretraining which increased the extraction accuracy[6]. There is also a challenge in understanding complex and specific medical language. By elaborating Large Language Models (LLM) that have been trained on large text collections, LLM can understand, generate, and process natural language contextually[7]. We also studied to extract nursing care records from nurse visit transcripts using specific prompts in LLM[8]. The results were quite promising for single-value output, but still not good enough for long sentence output that requires better context.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 2435-2445

We propose a low-code approach utilizing Flowise AI and Retrieval-Augmented Generation (RAG) methodology with Q&A output to address this issue. Flowise AI is a language model-based application development tool equipped with a visual interface, which allows users to intuitively design workflows through the manipulation of structured components[9]. The emergence of low-code and no-code platforms has changed the landscape of AI development, enabling broader access to AI technologies for users with varying levels of technical expertise. These platforms provide a visual development environment where users can design and implement AI-based solutions with minimal technical knowledge. With LLMs that can be implemented in low-code or even no-code, this is very beneficial for the wider community who may not have a computer science background but can still utilize this technology to solve problems they face, one of which is in the health sector. Research on low-code and no-code platforms shows their effectiveness in accelerating application development, increasing prototyping efficiency, and reducing reliance on specialized programming skills. Flowise AI stands out because it provides a visual interface for creating sophisticated language model-based applications with integrated components.

By utilizing the RAG methodology[10], the Q&A system allows for a harmonious combination of search and text generation capabilities, allowing more effective access to information from care records. The RAG methodology combines two main components in NLP: a retrieval module to search for relevant contextual information and a generative module that generates answers based on the acquired context[11]. The RAG model has gained popularity due to its capabilities in open-domain Q&A tasks by integrating a knowledge base into the model workflow. Studies have shown that RAG significantly improves answer accuracy in Q&A systems by linking generated responses to the retrieved context, reducing information errors, and increasing answer reliability[12]. Furthermore, RAG's flexibility in adapting to various domains makes it suitable for specific areas such as healthcare, where accurate information retrieval is critical for sound clinical decision-making.

Despite advances in information retrieval and NLP techniques, a gap exists in developing low-code AI systems that can handle domain-specific tasks such as information retrieval from care records. Low-code platforms such as Flowise AI, combined with the power of RAG, offer a promising solution by enabling the development of context-sensitive Q&A systems that are easily accessible to non-technical users. This study aims to fill this gap by leveraging Flowise AI to create a low-code environment where domain experts can easily design and implement Q&A systems for care records, thereby improving the efficiency and accuracy of information retrieval.

II. Research Methodology

A. Flowise AI Framework

Flowise AI is a low-code platform designed to simplify the development of Large Language Models (LLM) applications. With Flowise AI, developers can visually design workflows by intuitively connecting modular nodes, allowing for rapid application development without extensive coding. This framework supports a component-based approach, enabling users to assemble various modules like agents, caches, chains, conversational models, and more, to create tailored workflows.

One of the distinguishing features of Flowise AI is its integration with LangChain. LangChain is a library that facilitates chain management and agent-based processing for language models, making it an ideal companion for modular AI development[13]. By using Flowise AI, developers can leverage LangChain's capabilities to efficiently manage and process complex workflows within LLM applications. This integration enhances the platform's capacity for creating versatile and scalable language-based applications. In this research, Flowise AI is employed to structure a workflow consisting of multiple nodes:

- Recursive Character Text Splitter: This node breaks down text data into manageable chunks, enhancing processing efficiency.
- Pdf File: Used to upload and process PDF documents as source material.
- OpenAI Embeddings: This node generates vector embeddings for text data, enabling semantic search and similarity matching.
- ChatOpenAI: A conversational model based on OpenAI's language model, which interacts with users and generates responses.
- In-Memory Vector Store: A storage solution for embeddings that allows for quick retrieval during the query process.
- Conversational Retrieval QA Chain: An integrated component that facilitates question-answering by retrieving relevant data and generating contextually accurate responses.

Figure 1 shows the workflow connecting the nodes in Flowise AI. This workflow illustrates how each node is arranged and connected to create an efficient processing pipeline for applications based on LLM. Each node, including the Recursive Character Text Splitter, Pdf File, OpenAI Embeddings, ChatOpenAI, In-Memory Vector

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 2435-2445

Store, and Conversational Retrieval QA Chain, plays a specific role in this pipeline, from text segmentation to generating conversational responses based on retrieved information. With this configuration, Flowise AI enables the system to perform information retrieval and processing in a modular and structured manner.

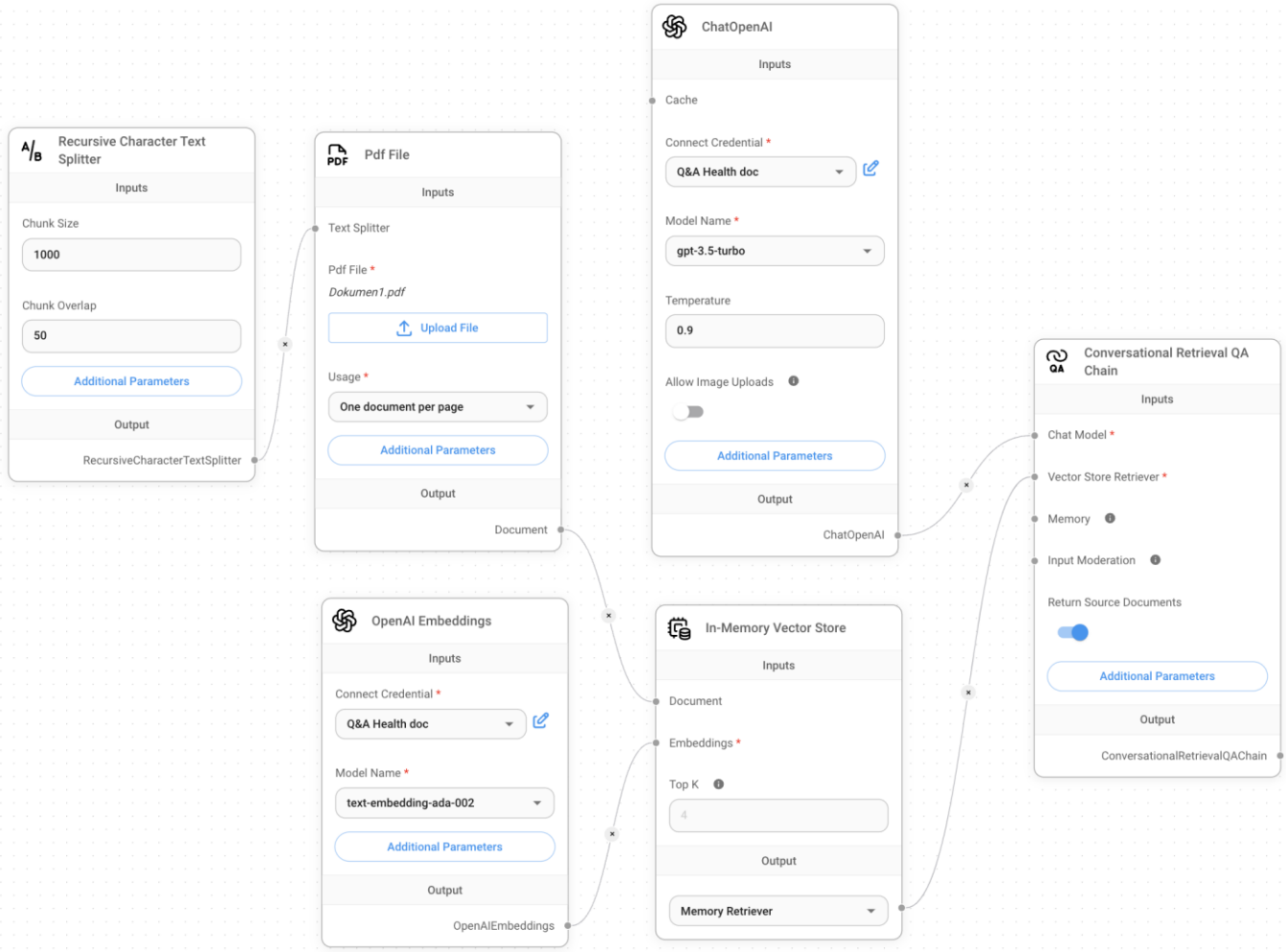


Figure 1. Flowise AI's workflow for connecting nodes

In the Recursive Character Text Splitter, setting the chunk size and chunk overlap is crucial to ensure that the text is divided into manageable pieces for further processing without losing context[14].

1. Chunk Size = 1000: By setting the chunk size to 1000 characters, we ensure that each text segment is large enough to carry meaningful context but not so large that it becomes difficult for the model to process. This size strikes a balance between providing sufficient context and maintaining processing efficiency. With 1000 characters, a large language model (LLM) can capture the essence of the information within a chunk without too much repetition or loss of context across different text segments.
2. Chunk Overlap = 50: An overlap of 50 characters is used to create a smooth transition between each chunk. This is important because when the text is divided, there is a risk that critical information might be located at the boundary between two chunks. By adding a 50-character overlap, we reduce the chance of losing key context or information that may be spread across consecutive chunks. This overlap helps maintain the narrative flow and context, resulting in more accurate and meaningful outputs when the text is processed by the model.

This configuration essentially provides a balance between processing efficiency and context integrity, allowing the system to retain accuracy in understanding and responding to the text.

The Pdf File component in the Flowise AI workflow plays a key role in handling and processing PDF documents as input sources. This component enables the system to ingest PDF files, extract their textual content, and convert it into a format suitable for further processing by the language model.

1. Text Extraction: The Pdf File component reads the contents of a PDF document, extracting text from each page. This step is crucial, as PDF files often contain complex layouts, including multiple columns, images, or embedded fonts, which require specialized handling to ensure accurate text extraction.
2. Content Structuring: Once the text is extracted, the Pdf File component helps in structuring it in a coherent format, breaking down the text into sections or paragraphs as required. This structuring prepares the text for

downstream components, such as chunking with the Recursive Character Text Splitter, enabling more efficient and accurate processing.

3. Versatility: This component allows the system to process a wide range of documents, making it useful for applications where knowledge sources are often stored in PDF format, such as research papers, reports, manuals, and legal documents. By supporting PDF inputs, Flowise AI can expand its reach to various document types, enhancing its adaptability across different domains.

In summary, the Pdf File component is essential for handling PDF-based content, converting complex document layouts into structured, plain text for seamless integration into the Flowise AI processing pipeline. This ensures that valuable information stored in PDFs can be effectively utilized in language model-based applications.

OpenAI Embeddings is a crucial component in workflows that require semantic understanding and similarity matching between text data[15]. When tuned with the text-embedding-ada-002 model, it enhances the efficiency and accuracy of generating embeddings-vector representations of text that capture semantic meaning. Here's an in-depth explanation of its functionality and advantages:

1. Purpose of OpenAI Embeddings: This node generates vector embeddings for text data. These embeddings are high-dimensional representations that capture the semantic essence of text, meaning that similar texts will have embeddings that are close to each other in the vector space. This capability allows the system to perform tasks like semantic search, similarity matching, and context-based retrieval, as it can find texts that are meaningfully related rather than simply matching keywords.
2. text-embedding-ada-002 Model: The text-embedding-ada-002 model by OpenAI is one of the most efficient and powerful embedding models available. It's known for its balance between performance and cost-effectiveness, providing high-quality embeddings while being computationally efficient. The model is trained to understand context and nuance in language, which improves the quality of semantic search and retrieval tasks.
3. Benefits of Using text-embedding-ada-002[15]:
   - Enhanced Semantic Understanding: The embeddings generated by text-embedding-ada-002 are rich in semantic information, enabling the system to capture subtle meanings, relationships, and nuances in the text.
   - Efficient Storage and Retrieval: Because the model generates compact and efficient embeddings, storing them in an In-Memory Vector Store is both memory-efficient and allows for quick retrieval.
   - Improved Retrieval-Augmented Generation (RAG) Performance: When integrated within a RAG framework, using text-embedding-ada-002 improves the quality of information retrieval[3]. This model ensures that the most relevant and contextually accurate data is retrieved, which supports the generation phase in providing precise and meaningful responses.
   - Flexibility Across Domains: Due to its general-purpose nature, text-embedding-ada-002 performs well across various domains and can be fine-tuned for specialized applications if necessary.

The In-Memory Vector Store is a storage solution that holds vector embeddings generated by models such as OpenAI Embeddings[16]. This component is essential for enabling quick retrieval of relevant information during the query process. Here's an explanation of its role and the purpose of tuning it with top K = 4:

1. Purpose of In-Memory Vector Store: The In-Memory Vector Store stores embeddings in a way that allows for fast and efficient retrieval. When a query or input text is received, the system searches this store to find the most relevant embeddings, which represent similar or contextually related pieces of information. Since these embeddings are stored in memory, retrieval is almost instantaneous, allowing the system to respond quickly to user queries.
2. How It Works: When a new query embedding is generated, the In-Memory Vector Store searches its stored embeddings to find the closest matches based on similarity measures, typically using cosine similarity or Euclidean distance. This search process identifies the embeddings that are closest to the query embedding in the vector space, implying a high degree of semantic similarity.
3. Tuning with Top K = 4: Setting top K = 4 means that the vector store will retrieve the top 4 most similar embeddings (or closest matches) to the query embedding. This tuning parameter impacts the retrieval process in the following ways:
   - Improved Accuracy: By retrieving the top 4 relevant embeddings, the system has access to multiple related pieces of information, increasing the chance of capturing the correct context or answer for the query.
   - Contextual Richness: With multiple relevant embeddings, the system can consider different aspects or perspectives related to the query, which can be especially useful in complex tasks like question-answering or document summarization.
   - Balanced Response: Limiting to 4 top results helps prevent information overload while ensuring that a sufficient number of relevant pieces of information are retrieved. This is particularly helpful in conversational AI or retrieval-augmented generation (RAG) applications, where providing too much information can dilute the response quality.

ChatOpenAI is a conversational model node based on OpenAI's language model, which generates responses by interacting with users based on provided prompts and contextual data. When configured with gpt-3.5-turbo and a temperature of 0.9, it creates a balance between generating coherent responses and introducing variability in the conversation[17]. Here's a breakdown of its functionality with these specific settings:

1. ChatOpenAI Node: The ChatOpenAI node is responsible for generating conversational responses in the workflow. It uses a language model to understand the input prompt or query and generates a response that is contextually relevant. This node can be integrated with other components, such as a retrieval mechanism, to answer questions based on specific information or to engage in interactive dialogues.
2. gpt-3.5-turbo Model: The gpt-3.5-turbo model is a variant of OpenAI's GPT-3.5, optimized for cost-effectiveness and conversational applications. Known for its improved speed and efficiency, gpt-3.5-turbo can handle complex language tasks and understand nuanced queries while maintaining high performance. This model is particularly suitable for chat-based interactions, as it is designed to engage in dialogues and generate responses that are natural and coherent.
3. Temperature = 0.9: The temperature parameter controls the randomness or creativity of the model's responses. When the temperature is set to 0.9, it introduces a higher degree of variability in the responses, making them more creative and diverse. Here's how this affects the ChatOpenAI node's behavior:
    - Enhanced Creativity: A temperature of 0.9 encourages the model to explore less predictable word choices and sentence structures, leading to more imaginative and varied responses. This is useful in scenarios where creativity is valued, such as brainstorming, storytelling, or generating responses that are not overly repetitive.
    - Balanced Coherence: While a temperature of 0.9 adds creativity, it still maintains coherence in responses, as the model doesn't stray too far from the core meaning of the input. This ensures that the answers remain relevant to the query or prompt.
    - User Engagement: With a slightly higher temperature, responses tend to be more engaging and dynamic, which can improve user experience in conversational applications where a rigid, robotic tone may not be desirable.

The Conversational Retrieval QA Chain is a critical component in the Flowise AI framework that enables a system to perform question-answering (QA) tasks with high contextual accuracy by combining conversational AI with retrieval-based processing[18]. This chain is designed to retrieve relevant information from a knowledge base and generate responses in a way that is both informative and contextually relevant to the user's query. Here's a detailed explanation of its functions and benefits:

1. Retrieval and Generation Integration: The Conversational Retrieval QA Chain combines two main processes-retrieval and generation to enhance response quality:
    - Retrieval Phase: In this phase, the system identifies and retrieves relevant information from an indexed knowledge base, which may include documents, PDF files, or other text sources. Using stored vector embeddings, the chain locates pieces of information that are semantically similar to the user's query, ensuring that responses are grounded in factual data.
    - Generation Phase: After relevant data is retrieved, a generative model like ChatOpenAI is used to create a coherent and contextually accurate response. The retrieved information serves as a foundation, guiding the generative model to produce answers that are both reliable and contextually relevant to the query.
2. Conversational Context Handling: This QA chain is optimized for maintaining conversational context. In multi-turn dialogues, it keeps track of previous interactions, allowing it to generate responses that reflect prior context. This feature is crucial in applications where users may follow up on previous questions or shift topics slightly, as it ensures continuity and coherence throughout the conversation.

B. RAG (Retrieval-Augmented Generation) Approach

The Retrieval-Augmented Generation (RAG) approach is a powerful framework that combines two main phases-retrieval and generation to produce accurate, contextually relevant answers. In the retrieval phase, the system searches for pertinent information from an existing knowledge base, which can include text documents, medical records, or other data sources. The retrieval module utilizes a trained model to select information most relevant to the query, allowing the system to answer questions based on factual data. In the generation phase, a generative model, such as GPT-3 or other large language models (LLMs), is used to produce responses based on the retrieved information, ensuring answers are both informative and contextually aligned with the question. This dual-phase approach helps RAG reduce inaccuracies and biases, enabling the system to respond with answers grounded in a broader context. In the Flowise AI workflow, the RAG process is implemented through several key nodes:

1. OpenAI Embeddings: This node is crucial to the retrieval phase in RAG. It generates vector embeddings for text data, allowing the system to perform semantic search and similarity matching[19]. By converting text into vector embeddings, this node enables the system to locate relevant information based on the context of the query, rather than relying solely on keyword matching. This is particularly beneficial for applications such as conversational AI, document search, and question-answering, where accurate context matching is essential.
2. In-Memory Vector Store: This node also operates within the retrieval phase, storing the embeddings generated by the OpenAI Embeddings node. During the query process, it enables fast retrieval of relevant information. By maintaining an accessible store of embeddings, the In-Memory Vector Store ensures that pertinent data can be quickly accessed, which is critical for the efficiency of retrieval in the RAG framework.
3. Conversational Retrieval QA Chain: This node represents the core of the RAG approach, integrating both retrieval and generation phases. In the retrieval phase, it searches the In-Memory Vector Store for the most relevant data. During the generation phase, it uses this retrieved information to generate accurate, context-aware answers with the help of a language model like ChatOpenAI. This chain facilitates the seamless transition between retrieving relevant information and generating responses, ensuring that answers are grounded in context.

In the RAG setup, the text-embedding-ada-002 model plays a critical role in the retrieval phase by generating embeddings that capture the contextual meaning of queries and documents. This model enables the system to identify the most relevant information based on context, making it well-suited for scenarios requiring high contextual relevance, such as question-answering systems.

The RAG framework also incorporates a top K = 4 configuration for retrieving the four most relevant embeddings, which supports the generation phase by providing a small, focused set of contextually appropriate data[20]. This subset of information serves as the basis for the language model to produce responses that are concise, relevant, and accurate, enhancing the system's reliability.

By using RAG in the Conversational Retrieval QA Chain, Flowise AI reduces the likelihood of generating inaccurate or fabricated answers, commonly known as hallucinations. The grounding of responses in actual knowledge base content enhances accuracy and ensures that responses are contextually relevant to both the immediate question and the ongoing conversation. This approach is particularly advantageous in fields that demand high precision, such as healthcare and customer support, where reliable, contextually relevant answers are critical.

C. Dataset

In this study, we utilized nursing documentation files from the Ministry of Health of the Republic of Indonesia, dated 2017, as the primary data source[21]. These files contain various examples of nursing assessment report forms in diverse formats, including narrative sentence reports, structured tables, checklists for assessment elements, and fill-in-the-blank fields separated by colons. This variety in formats presents a unique challenge in data processing, particularly in ensuring consistency and accuracy when converting information from these different formats.

The dataset derived from these documents serves as the ground truth for this study, where we conduct comparative analysis to assess the precision and consistency of information generated through automated question-answering. Our Retrieval-Augmented Generation (RAG) system leverages this dataset as a knowledge source, allowing us to determine whether the information generated through Q&A aligns with the information contained within the dataset. This approach is crucial for evaluating the model's ability to reference accurate information, supporting the need for precise nursing documentation. By using this dataset as both the ground truth and a RAG data source, we aim to measure the system's capacity to capture relevant and accurate information and to evaluate how well the model can generate responses that align with the reference documents. This is expected to contribute to the development of a reliable AI-based decision support system in the field of nursing.

III. RESULT AND DISCUSSION

In this section, we evaluate the effectiveness of the proposed Q&A system developed using Flowise AI with Retrieval-Augmented Generation (RAG) in accurately retrieving information from nursing care records and aligning it with ground truth data. The primary objective is to demonstrate that responses generated by the Q&A system accurately reflect the content in the source nursing documentation, even when data is presented in various formats, including tables, narrative sentences, checklists, and fill-in-the-blank fields.

The system was tested by comparing Q&A outputs with the ground truth from nursing records provided by the Ministry of Health of the Republic of Indonesia (2017). For each query, the retrieved and generated answers were assessed to determine if they matched the relevant details within the ground truth documents. The results indicate

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 2435-2445

a high accuracy rate, showing that the system can handle multiple data formats and accurately retrieve contextually appropriate information.

- Tables: When questions related to tabular data were posed, the system was able to extract specific values and fields, interpreting structured information correctly. For example, when asked about vital signs recorded in a tabular format, the Q&A system accurately retrieved values. Fig. 2 below shows an example of Q&A about medical history related to hepatitis C:

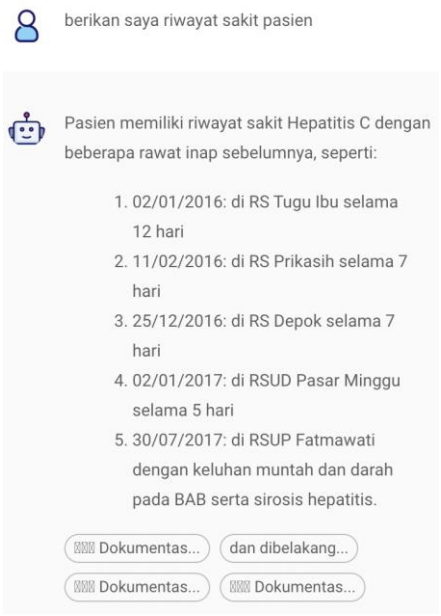


Figure 2. Q&A on medical history

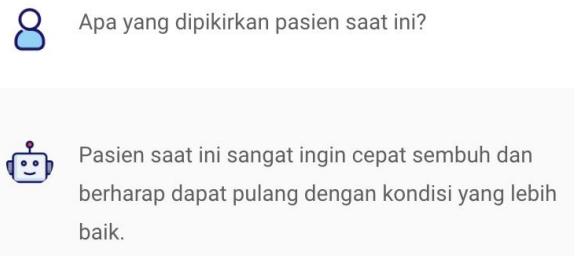
In order to evaluate the Q&A results, we compare the Q&A results with ground truth contained in a table in the report. The ground truth image can be seen in Figure 3 below.

c. Riwayat dirawat di Rumah sakit (kapan, alasan dan berapa lama) :

No	Tanggal	Alasan	Berapalama	Tempat
1	02/01/2016	Hepatitis c	12 hari	RS Tugu ibu
2	11/02/2016	Hepatitis c	7 hari	RS Prikasih
3	25/12/2016	Hepatitis c	7 hari	RS Depok
4	02/01/2017	Hepatitis c	5 hari	RSUD Pasar minggu
5	30/07/2017	Muntah dan BAB darah dan sirosis hepatis	sampai sekarang	RSUP Fatmawati

Figure 3. Ground Truth: Data in the form of a table in the report form as a reference for evaluating Q&A results

- Narrative Sentences: In cases where the data was presented in sentence form, the system displayed strong contextual understanding, accurately retrieving and generating responses that matched the sentence-based records. For instance, when queried about specific nursing actions or observations described narratively, the system produced answers closely aligned with the original sentences. Figure 4 shows the Q&A results used to gather information about what the patient is currently thinking.



JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 2435-2445

Figure 4. Q&A Results for patient's current thoughts

As shown in Figure 5, ground truth is presented as narrative sentences. This format is used as a reference to ensure conformity between the Q&A results and the expected information.

f. Persepsi pasien terhadap penyakitnya
- Hal apakah yang sangat dipikirkan saat ini ? ingin cepat sembuh
- Apakah harapan setelah menjalani perawatan ? cepat sembuh total

Figure 5. Ground Truth in narrative sentence format for evaluating Q&A consistency

- Checklists: Due to their concise format, checklist items pose unique challenges. However, the Q&A system demonstrated the ability to interpret and match checklist entries effectively. When asked about a specific checklist item, such as “hearing function”, as in Figure 6, the system successfully retrieved the appropriate checklist item and provided an accurate response.

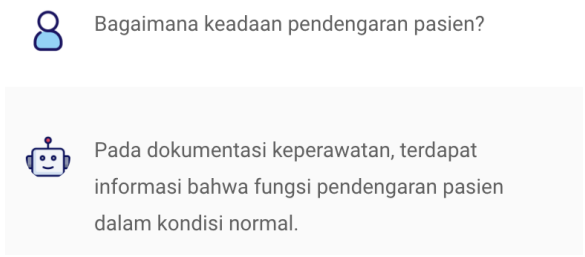


Figure 6. Q&A results for patient’s hearing function

Figure 7 demonstrates that the generated Q&A is able to extract data from the checklist format. As shown in the image, the extracted results match exactly with the information provided in the checklist, indicating the system's ability to understand and extract information from the checklist format accurately.

f. Tintus	: (–) Ya	(√) Tidak
g. Fungsi pendengaran	: (√) Normal	(–) Kurang(–) Tuli
h. Gangguan keseimbangan	: (√) Tidak	(–) Ya
i. Pemakaian alat bantu	: (–) Ya	(√) Tidak

Figure 7. Ground truth in checklist format

- Fill-in-the-blank Columns: The model also proved capable of interpreting and extracting data from fill-in-the-blank sections as shown in Figures 7 and 8. By focusing on the context provided by the surrounding text, the system was able to match answers to relevant columns, demonstrating flexibility in handling a variety of documentation structures.

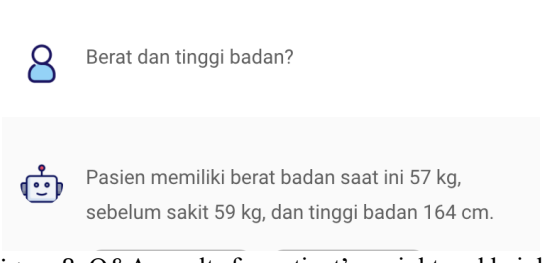


Figure 8. Q&A results for patient’s weight and height

Pemeriksaan Fisik Umum:

a. Berat badan	: 57 kg (saat ini), 59 kg (sebelum sakit)	
b. Tinggi badan	: 164 cm	
c. IMT	: Tidak dapat di hitung	Berat badan ideal : 57.6-70.4 kg
d. TTV	: Tekanan darah :100/70 mmHg	Nadi :128 x/menit
	Frekuensi nafas :22x/menit	Suhu tubuh : 36.3°C
e. Lingkar lengan	: 24 cm	
f. TSF	: 7 cm	
g. Keadaan umum	: (–) Ringan (√) Sedang	(–) Berat
h. Pembesaran kelenjar getah bening: (√) Tidak (–) Ya, lokasi: Tidak ada		

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 2435-2445

Figure 9. Ground truth in fill-in-the -blank-field

We designed 53 questions based on the information available in the form to analyze the effectiveness of the developed model. These questions cover various important aspects of the data to be evaluated. From the test results, the model provided answers that were in accordance with the ground truth on 39 questions, indicating a fairly high level of accuracy in capturing the correct information. However, there were 4 answers that were not in accordance with the ground truth, indicating an error in the interpretation or processing of information by the model. In addition, the model also provided 10 answers stating that the requested information was not in the form. This could be due to the limitations of the form in providing relevant data for a particular question or the model's ability to identify missing information. The distribution of these results provides an overview of the strengths and weaknesses of the model in extracting information from the form, as well as being the basis for further improvement steps. A more detailed explanation of the results can be seen in Figure 10.

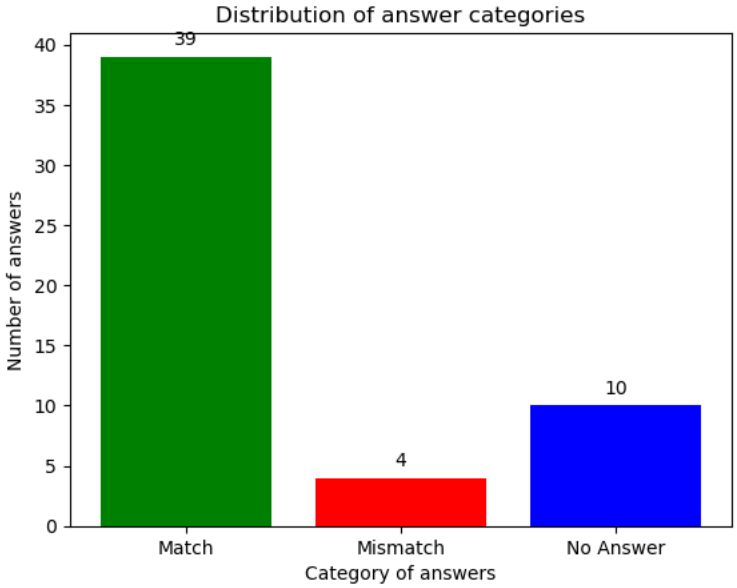


Figure 9. Disribution of answer categories

The graph in this figure illustrates the proportion of each category, namely the number of answers that match, mismatch, and no answer. The graph shows that most answers come from the "Match" category, which indicates the model's potential to capture relevant information. However, the "Mismatch" and "No Answer" categories are important points of attention that can be used to further analyze the causes of errors and identify future improvement opportunities in the model.

Traditional information retrieval methods, such as manual human search or rule-based systems, have a number of significant drawbacks. While generally accurate when performed by experienced individuals, manual searches take significantly longer to complete the same number of queries. They are prone to errors, especially when dealing with complex or unstructured forms. On the other hand, rule-based systems rely heavily on predefined rules, making them less flexible in handling new queries or data variations. In addition, rule-based systems tend to be less accurate, with common errors being the inability to understand complex contexts and the tendency to provide speculative answers when the requested information is not found.

Furthermore, studies that have used LLM to generate specific prompts also have limitations. This approach is still limited to the prompts given to obtain the desired output, making it less adaptive in handling a wider variety of contexts. In addition, the results produced by LLM tend to be weak when asked to provide output in the form of complex sentences. In fact, in some cases, the model still provides speculative answers if it does not find relevant information in the form provided. This shows that although LLM has great potential, its use for information retrieval still requires further development to improve its accuracy and reliability.

The proposed system offers significant advantages. It is more efficient and excels at identifying correct answers without giving speculative responses when information is missing. This capability, which is lacking in previous methods, reflects the system's strength in detecting and recognizing the absence of information. Additionally, the system is designed to be more adaptive, allowing the processing of new questions or varying data without the need for manual rule changes. These advantages make the proposed system a more reliable and flexible solution compared to traditional approaches.

A total of 53 questions were tested, and four mismatched answers were found, as well as 10 no-answer outputs, which were the primary focus of our analysis. One challenge that emerged was the system's difficulty in understanding the context of more complex questions, especially when the data on the form is unstructured or has many similar elements. This condition causes errors in matching answers to the ground truth, even though relevant information is actually available. For example, on the question "What about the patient's skin color?", the model gives the answer "normal", while the correct ground truth is "pale". The answer "normal" may be indicated by other skin condition information, such as good skin turgor, good skin condition, and no skin abnormalities, which should not replace more specific information about the patient's skin color. This challenge indicates the need to improve the system's ability to understand the semantic relationship between questions and data on the form, so that the system can provide more accurate and relevant answers.

The implementation of Flowise AI in healthcare requires attention to several important factors, especially infrastructure and user training. In order to run AI models, large and secure data storage as well as powerful servers are needed. In addition, the integration of this system with existing Electronic Medical Records (EHR) is essential to ensure smooth and efficient data flow. Furthermore, user training is equally important. Medical personnel, including doctors, nurses, and administrative staff, need to be trained to be able to utilize Flowise AI optimally. This training includes a basic understanding of how to interact with the system, as well as how to handle cases that cannot be handled by the system. In addition, users must also be taught about the importance of maintaining patient data privacy and understanding existing security policies. Without proper training, the adoption of this technology can be hampered, even though the system itself is designed to make their jobs easier.

This study suggests several research directions that can be pursued to improve the performance of the proposed system, especially when retrieving information from medical forms. For instance, the development of more sophisticated models can improve the system's ability to understand the context and semantics of unstructured or similar data. In addition, the system's ability to detect missing information still needs to be improved, with research exploring more accurate missing detection techniques using machine learning.

Research can also focus on integrating the system with Electronic Medical Records (EHR), allowing the system to pull data directly from patient medical records, as well as reducing the reliance on manual forms that are often unstructured. Thus, the system can be more effective in providing relevant and accurate answers. On the other hand, much medical data is presented in unstructured formats, such as doctor's notes or patient interview transcripts, which require further development in NLP to extract information more efficiently.

Finally, research can also focus on improving the robustness of models to unexpected variations in questions and data formats. A more flexible and adaptive system, capable of handling different types of input, be it text, numbers, or images, would be very useful to improve the performance of the system in real-world scenarios. Using transfer learning-based approaches or fine-tuning models for different types of data and question formats can be an effective step in addressing this challenge. Research in these areas will enable the development of more robust and reliable systems in various healthcare contexts, thereby improving the accuracy and efficiency of medical information retrieval.

IV. CONCLUSION

This study demonstrates the effectiveness of a low-code approach using Flowise AI with Retrieval-Augmented Generation (RAG) to build a Q&A system for extracting information from diverse nursing documentation formats. By leveraging the capabilities of Flowise AI, integrated with LangChain and OpenAI's LLM models, the system was able to accurately retrieve information from various document structures, including tables, narrative sentences, checklists, and fill-in-the-blank fields. The results highlight the system's ability to handle diverse data formats, ensuring consistency and reliability in retrieving critical healthcare information.

The RAG methodology, which combines retrieval and generation phases, proves essential in maintaining contextual accuracy and enhancing answer reliability. The use of OpenAI Embeddings and an In-Memory Vector Store contributes to the system's strong semantic search capabilities, allowing it to perform well even in complex queries that require contextual understanding. This low-code solution demonstrates the potential for expanding AI accessibility to non-technical users, particularly in the healthcare field, where accurate and timely information retrieval can significantly impact patient outcomes.

In summary, the low-code Q&A system developed in this research offers a promising solution for enhancing healthcare documentation retrieval, providing healthcare professionals with a scalable and accessible tool to improve patient care. This study lays the foundation for further research and development in low-code AI solutions, particularly for domain-specific tasks in healthcare and other industries requiring high precision in information retrieval.

REFERENCES

[1] Hamdhana D, Kaneko H, Victorino JN, Inoue S. Improved Evaluation Metrics for Sentence Suggestions in Nursing and Elderly Care Record Applications. *Healthcare*. 2024; 12(3):367. https://doi.org/10.3390/healthcare12030367
[2] Tsai, C. H., Eghdam, A., Davoody, N., Wright, G., Flowerday, S., & Koch, S. (2020). Effects of electronic health record implementation and barriers to adoption and use: a scoping review and qualitative analysis of the content. *Life*, *10*(12), 327.
[3] Badawy, M., Ramadan, N., & Hefny, H. A. (2023). Healthcare predictive analytics using machine learning and deep learning techniques: a survey. *Journal of Electrical Systems and Information Technology*, *10*(1), 40.
[4] De Benedictis, A., Lettieri, E., Gastaldi, L., Masella, C., Urgu, A., & Tartaglini, D. (2020). Electronic Medical Records implementation in hospital: An empirical investigation of individual and organizational determinants. *PloS one*, *15*(6), e0234108.
[5] Lamy, M., Pereira, R., Ferreira, J. C., Vasconcelos, J. B., Melo, F., & Velez, I. (2019). Extracting clinical information from electronic medical records. In *Ambient Intelligence–Software and Applications–, 9th International Symposium on Ambient Intelligence* (pp. 113-120). Springer International Publishing.
[6] Rybinski, M., Dai, X., Singh, S., Karimi, S., & Nguyen, A. (2021). Extracting family history information from electronic health records: natural language processing analysis. *JMIR Medical Informatics*, *9*(4), e24020.
[7] Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... & Mian, A. (2023). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.
[8] Hamdhana, D., Harada, K., Oshita, H., Sakashita, S., & Inoue, S. (2024). Toward Extracting Care Records from Transcriptions of Visiting Nurses Using Large Language Models. *International Journal of Activity and Behavior Computing*, *2024*(2), 1-17.
[9] Gao, L., Liu, H., & Huang, H. (2021). Low-code platforms for developing artificial intelligence applications. *Journal of Applied Computing and Information Technology*, *20*(2), 12-20.
[10] Lewis, P., Oguz, B., Rinott, R., Riedel, S., & Stoyanov, V. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*.
[11] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.
[12] Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., ... & Li, Q. (2024, August). A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 6491-6501).
[13] Asyrofi, R., Dewi, M. R., Lutfhi, M. I., & Wibowo, P. (2023, August). Systematic Literature Review Langchain Proposed. In *2023 International Electronics Symposium (IES)* (pp. 533-537). IEEE.
[14] Chen, Y., Li, Y., Ding, B., & Zhou, J. (2024). On the Design and Analysis of LLM-Based Algorithms. *arXiv preprint arXiv:2407.14788*.
[15] Xian, J., Teofili, T., Pradeep, R., & Lin, J. (2024, March). Vector search with OpenAI embeddings: Lucene is all you need. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* (pp. 1090-1093).
[16] Liang, S., Wang, Y., Yuan, Z., Liu, C., Li, H., & Li, X. (2022, July). VStore: in-storage graph-based vector search accelerator. In *Proceedings of the 59th ACM/IEEE Design Automation Conference* (pp. 997-1002).
[17] Lo, L. S. (2023). The CLEAR path: A framework for enhancing information literacy through prompt engineering. *The Journal of Academic Librarianship*, *49*(4), 102720.
[18] FlowiseAI(2024, Oct). FlowiseAI: Conversational Retrieval QA Chain. [Online]. Available: https://docs.flowiseai.com/integrations/langchain/chains/conversational-retrieval-qa-chain
[19] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459-9474.
[20] Yu, Y., Ping, W., Liu, Z., Wang, B., You, J., Zhang, C., ... & Catanzaro, B. (2024). Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *arXiv preprint arXiv:2407.02485*.
[21] Dinarti, D., & Yuli Mulyanti, Y. (2017, Oct). Dokumentasi keperawatan. [Online]. Available: https://poltekkesbanten.ac.id/wp-content/uploads/2017/12/PRAKTIKA-DOKUMEN-KEPERAWATAN-DAFIS.pdf