Retrieval Augmented Generation: An Evaluation of RAG-based Chatbot for Customer Support

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Data Analytics
June 2024
Dishant Sukhwal

UNIVERSITY OF TURKU
Department of Computing

DISHANT SUKHWAL: Retrieval Augmented Generation: An Evaluation of RAG-based Chatbot for Customer Support

Master of Science (Tech) Thesis, 55 p.
Data Analytics
June 2024

The rapid advancements in generative artificial intelligence and large language models have revolutionized the field of natural language processing, giving rise to sophisticated frameworks such as Retrieval Augmentation Generation (RAG). The RAG framework combines the strengths of information retrieval and text generation, enabling the development of highly capable chatbots. As chatbots become increasingly popular in various domains, including customer support, their ability to deliver accurate and relevant responses is crucial. The evaluation of these chatbots is equally important to ensure their reliability and effectiveness. This thesis evaluates the performance of a Retrieval Augmentation Generation (RAG)-based chatbot for customer support developed at a software company. This study focus specifically on the evaluation of the retriever module. The study investigates the impact of keyword generation/extraction, the effects of different prompting techniques, and additional parameters influencing system performance. A dataset for evaluation was created as a part of this study, encompassing a wide range of customer queries related to four different software products offered by the company.

Generative AI was utilized as a language assistant to ensure language accuracy, recommend improved expressions, and assist in structuring the text throughout the thesis.

Keywords: Retrieval Augmentation Generation, RAG, evaluation, LLM, chatbot, Information retrieval

Contents

1 Introduction

1.1 Background

Recent advancements in the field of Generative AI have accelerated the adoption of AI-based chatbots, particularly in the E-commerce industry. Large language models such as GPT-3 and GPT-4, have not only surpassed the traditional language comprehension and generation but have also reached a level of imagination and creativity comparable to that of humans [1]. Businesses are increasingly employing AI-driven chatbots for customer support and assistance as they significantly impact the way businesses engage with their customers. Incorporating chatbots in customer support provides a plethora of benefits, including 24/7 availability, quick response times, reduced costs and scalability [2] [3]. These chatbots are particularly beneficial for large organizations that have thousands of customers all over the world.

While large pre-trained language models demonstrate strong performance across a wide range of NLP tasks including question answering, they demonstrate subpar performance in knowledge-intensive tasks, for example, when asked questions regarding a specific product or service offered by a company. The language model struggles to generate a useful response as it lacks exposure to the product or service-specific information during its training [4] [5]. As a promising solution to overcome this challenge, the Retrieval Augmentation Generation, or RAG, framework was introduced

by Facebook (now Meta) in 2020. The main components of the RAG framework are a retriever and a generator. As the name suggests, the retriever is responsible for fetching information from an external source based on the input query. The generator, which is a large pre-trained language model, is responsible to generate the response by extracting information from the documents retrieved by the retriever [6] [4].

1.2 Company Overview

The research presented in this thesis benefited from collaboration with an industry partner. For privacy reasons, the names of the company and products involved in this collaboration have been withheld from disclosure. The company is a global leader in providing technology solutions for a diverse range of industries, including construction, agriculture, transportation, and geospatial sectors. The company's leading software product 'Product 1', a software solution for structural engineering is revolutionizing the way construction projects are planned, designed and executed using its Building Information Modeling (BIM) capabilities. Alongside 'Product 1', the company offers additional software products designed to enhance construction information workflows. This study includes four different products in total. These products are available globally, serving customers across multiple countries. Consequently, the customer support team receives substantial amount of customer queries on a regular basis.

System Context

A RAG-based chatbot has been implemented as a proof of concept to facilitate customer interactions across various software products. It is capable of addressing customer queries, providing product information and helping through the troubleshooting process. Initially, the user selects the software product related to their

query. The user can also select a specific product version, by default the chatbot generates answers which are not specific to any particular product version. Additionally, the user has the option to select a particular product environment. While the product field is mandatory, providing a version and environment is optional. In the next step, the user submits the actual query. Once the system receives a query, it first extracts keywords from the actual query. The generated keywords are then used to perform the search and fetch top five relevant documents from the document store. Finally, the large language model is augmented with the top five relevant documents along with the user query, and it is instructed to produce an answer for the query based on the provided documents.

1.3 Problem Statement

Language models, such as ChatGPT, can answer basic questions regarding a product or service offered by a company. However, the model faces difficulty in generating precise and contextually relevant information when prompted with complex queries related to a specific product or service. A RAG-based chatbot can address these issues and enhance the performance by improving content relevancy, better handling of out of domain requests and reducing the risk of misinformation. At the same time, establishing an effective method for evaluating its performance is crucial. A good evaluation protocol can help identify any limitations or areas for improvement in the current system, such as gaps in knowledge retrieval.

1.4 Objectives

This thesis aims to evaluate the performance of a Retrieval Augmentation Generation (RAG) based chatbot for customer support in generating contextually relevant and accurate responses across various software products offered by the company

with varying complexities of queries. This study has a specific focus on assessing the performance and effectiveness of the retriever component. A high-quality dataset is crucial for successful evaluation of any system as it directly influences the effectiveness of the study, therefore, the primary objective of this study is to create a dataset containing diverse customer queries with relevant context. The dataset will serve as foundational resource for the experiments conducted during this study.

Another aspect of the study is determining the impact on performance by incorporating the keyword extraction step prior to retrieving the documents, as opposed to completely bypassing the keyword extraction process. The experiments conducted during the study aims to compare the performance of retrieval with keywords generated/extracted by using different prompts. This investigation will provide insights into the effectiveness of various prompts as well as understanding of how different input strategies influence the chatbot's performance. Furthermore, this study seeks to discover any other parameter that can be modified to improve the quality of the generated response.

2 Literature Review

2.1 Information Retrieval

Information retrieval has witnessed significant advancements over the past decade. Over time, the traditional methods of information retrieval have been replaced by a question-answering approach. Previously, an information retrieval system was expected to find all potential resources containing matching terms to the user search query. Today, these systems have evolved to not only retrieve information but also analyze it to extract the precise answer to the user question. Advanced models are designed to filter and rank documents within a vector store when prompted by a query. These models use various informational retrieval and search techniques. The fundamental concept is to transform the whole document set as well as the incoming query into a common semantic encoding space. The encoded query is compared against the encoded document set and the top matching documents can be retrieved [7]. The most crucial component of retrieval-based models is to create the shared semantic encoding space. The most frequently employed methods for information retrieval are:

2.1.1 TF-IDF

Term Frequency - Inverse Document Frequency or TF-IDF, is one of the traditional information retrieval methods based on the informativeness or importance of a par-

ticular term in a corpus. It is a combination of Term Frequency, which measures how frequently a word appears in a particular document, and the Inverse Document Frequency, which is the inverse of the number of documents in which the term appears. The central principle behind this method is that not all words are equally important in a document, so TF-IDF assigns weights based on the importance of the word in that document. For instance, common words such as articles (e.g., "a," "an," "the") and prepositions (e.g., "in," "of") frequently appear in a document, however, they are less significant. On the other hand, infrequent/rare words carry greater significance. The importance of a word is calculated as the product of TF and IDF [8]. Given a document set D, a term t and a document $d \in D$.

$$TF_{(t,d)} = \frac{n_{(t,d)}}{|d|} \tag{2.1}$$

$n_{(t,d)}$ represents the number of occurrences of term t in an individual document d and $|d|$ represents the total number of terms in the document d

$$IDF_{(t,D)} = \log\left(\frac{|D|}{m_{(t,D)}}\right) \tag{2.2}$$

$|D|$ represents the total number of documents and $m_{(t,D)}$ is the number of documents in which the term t appear.

$$W_t = \frac{n_{(t,d)}}{|d|} \times \log\left(\frac{|D|}{m_{(t,D)}}\right) \tag{2.3}$$

Equation 2.3 is adapted from J. Ramos [8].

W_t is the weight assigned to the term t and indicates its importance. A higher value of W_t indicates that the term is rare and holds more importance. Conversely, a lower value indicates that the term is common and less important. While TF-IDF is a simple, straightforward and effective method for information retrieval, this

approach fails to capture the semantic relationship in text, especially synonyms and polysemies [9]. Additionally, TF-IDF uniformly increases the importance of a term based on its frequency in the document, which may not always be accurate [10].

2.1.2 BM25

Best match 25 or BM25, is another famous information retrieval method. It is a probabilistic model and an extension of TF-IDF designed to address its limitations. In addition to Term Frequency and Inverse Document Frequency, BM25 also considers Term Saturation and Document Length Normalization. With term saturation, as the frequency of an individual term increases, the contribution to the weight (W_t) by each successive occurrence of that term diminishes (in the case of TF-IDF the contribution is uniform). This enables the algorithm to assign balanced weights/scores to frequently occurring terms. This is achieved by introducing a new parameter k, that controls the degree of saturation. The value of k is usually set to be between 1.2 and 2 [10] [11]. Equation 2.4 is credited to I. Heggo and N. Abdelbaki [8].

$$\frac{TF_{(t,d)}.(k+1)}{TF_{(t,d)}+k} \tag{2.4}$$

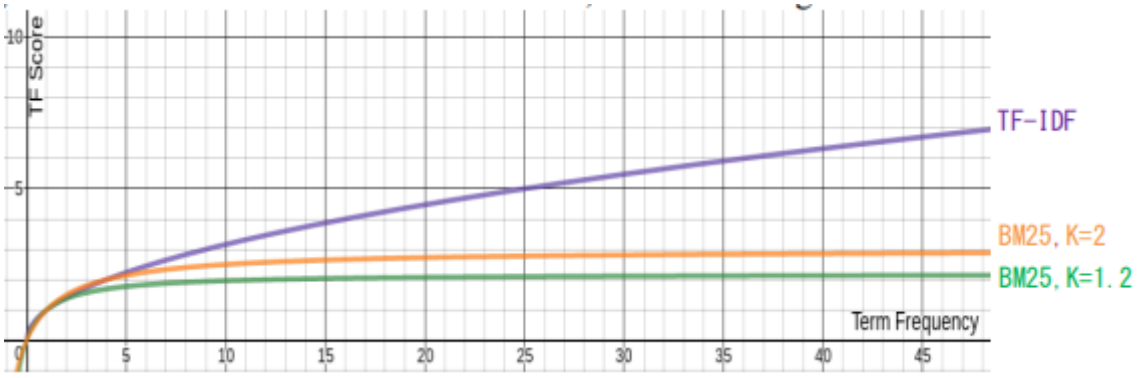

Figure 2.1: *The figure compares the TF score of TF-IDF and BM25 (with k = 1.2 and 2). This figure has been used from I. Heggo and N. Abdelbaki [8]*

In addition to term frequency, the length of the document is another crucial factor that influences the search results. A long document is likely to have higher number of matching terms as opposed to a shorter document. The reason for longer document lengths can be attributed to higher verbosity (more words) or scope (more contents). This might result in the algorithm giving higher preference to longer documents. Document length normalization is therefore an important step to reduce this bias introduced by varying document lengths. A normalization factor should be added to penalize the documents with longer lengths due to high verbosity, at the same time it should also avoid over-penalization of longer documents due to high scope [11]. This is a achieved by defining the normalization factor (B) with relation to the average document length as follows:

$$B = \left((1-b) + b\frac{|d|}{avdl}\right) \quad \text{where} \quad b \in [0,1] \tag{2.5}$$

Equation 2.5 is adapted from Stephen E. Robertson and Hugo Zaragoza [11]. $b = 1$ would lead to full normalization and $b = 0$ would mean no normalization.

2.1.3 Vector Space Model

The Vector Space Model (VSM) represents the documents and queries numerically in the form of vectors in a multi-dimensional space, where each term is a dimension. This representation of documents as vectors allows computation of similarity between document vectors and the query vector based on certain similarity measure. A vocabulary is created by collecting all the unique terms present in the corpus of documents. After establishing the vocabulary, each document is represented as a vector of term-document matrix, where rows correspond to terms in the vocabulary and columns correspond to the documents. Since the size of the vocabulary can be huge, the document vectors can be high-dimensional and extremely sparse. Each term in the document vector can be assigned a weight that denotes the importance of the corresponding term in the document. Similarly, queries are also represented as vectors in a high-dimensional space [12]. One of the most commonly used similarity measures is Cosine similarity due to its simplicity in computation and superior performance compared to alternative measures such as Jaccard measure and Euclidean measure. The Cosine similarity of a document vector d_i and query vector q is computed as the normalized dot product of the two vectors:

$$\cos(d_i, q) = \frac{d_i \cdot q}{\|d_i\| \cdot \|q\|} \tag{2.6}$$

It computes the cosine of the angle between the query and document vectors irrespective of their magnitudes. The cosine function has a range of [-1,1], however, all the vectors are in the first quadrant, since the word count is non-negative, the range is [0,1] [13].

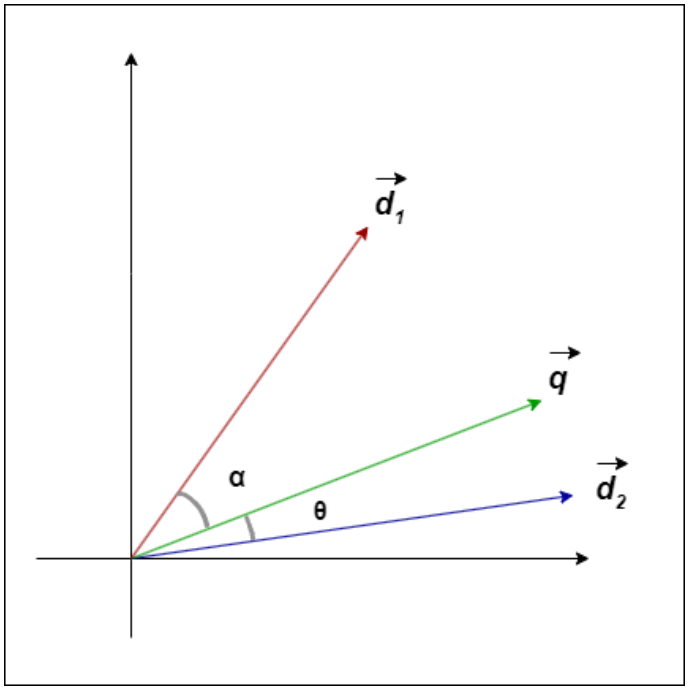

Figure 2.2: *Example of cosine similarity between query and document vectors in 2D, q is more similar to d_2 as opposed to d_1*

Smaller angle between two vectors indicates higher similarity (cosine value is closer to 1). Conversely, larger angle between two vectors indicates lower similarity (cosine value is closer to 0).

2.2 RAG Framework

RAG Framework has become a promising solution to develop chatbots that require specialized knowledge to generate useful responses. The framework combines the power of both retrieval models and generative large language models resulting in a significant increase in the quality and relevance of generated response. Additionally, it also eliminates the necessity to train the large language model on a specific corpus which requires a substantial amount of time and incurs significant costs. Integrating the retriever and generator overcomes the inherent limitations of both components.

The chatbot's capability to generate useful responses relies more on the retriever and to a lesser extent on the generator. Due to significant advancements in recent years, the generator (or the large language model) has nearly perfected its performance in extracting the required information given a context and generating answers based on it.

Three primary components of RAG Framework are: the Knowledge Base, the Retriever Model and the Generator Model.

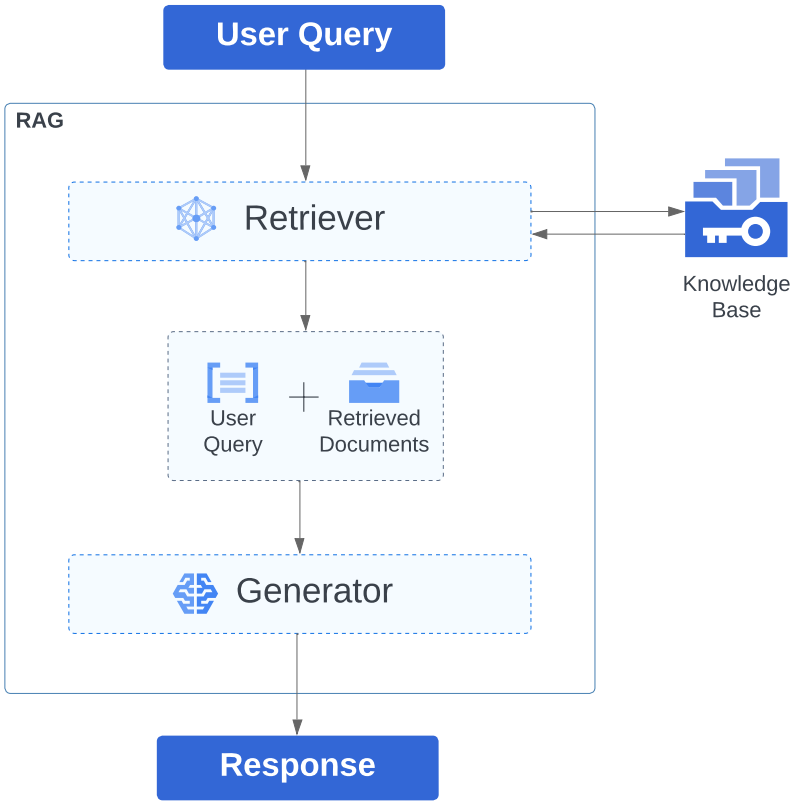

Figure 2.3: Retrieval Augmented Generation (RAG) Architecture.

2.2.1 Retrieval

Sparse-vector Retrieval

The goal of sparse-vector retrieval is to retrieve relevant examples from a collection of documents by comparing the input sequence with each document in the collection. It is called "sparse-vector" because the majority of elements in the document vector are zeros. The documents are fetched based on a certain similarity measure, two most commonly used similarity measures are TF-IDF and BM25 [14].

Dense-vector Retrieval

In contrast to sparse-vector retrieval, which focuses on matching exact keywords, dense-vector retrieval focuses on finding related examples by understanding the semantic meaning of the text. A dense-vector represents the meaning of the text rather than the words/terms themselves. Pre-trained language models are used to convert text into low-dimensional numerical vectors and the similarity is calculated using the inner product of the vectors [14].

Task-specific Retrieval

A task-specific retrieval system learns from the data to search for the best matching information to help with the task at hand. In this approach, the memory retriever is integrated with the downstream generation model. Finally, both the memory retriever and the generation component are trained together to optimize performance for a specific task [14].

Augmenting the pre-trained language model with relevant context significantly improves performance. Moreover, employing basic information retrieval techniques can yield outcomes comparable to those of a supervised DrQA system. As highlighted by Petroni et al. in their paper [15], *"We demonstrated a simple technique*

to greatly improve factual unsupervised cloze QA by providing context documents as additional inputs. We used oracle documents to establish an upper bound to this improvement, and found that using off-the-shelf information retrieval is sufficient to achieve performance on par with the supervised DrQA system."

2.2.2 Generation

The introduction of transformers has marked a revolutionary shift in natural language processing, particularly in the domains of language comprehension and generation. Attention mechanism is a method that enables the neural network to focus on important parts of the text. This is achieved by assigning higher weights to important parts of the text, enhancing its ability to understand and generate text. By replacing the recurrent layers with attention layers, the transformer gains the ability to capture long-range dependencies and it significantly reduces the training time as the tokens can be processed in parallel [16]. A modified version of transformer, Generative Pre-Training (GPT), consists of a stack of transformer decoder layers. The GPT model is pre-trained on a huge corpus of text in an unsupervised manner and then fine-tuned for downstream tasks using supervised learning. These models have achieved state-of-the-art performance across a wide range of NLP tasks including question answering [17].

Challenges faced by LLMs

LLMs have achieved remarkable performance in various natural language processing tasks, however, they still face several limitations and challenges:

Out of Date Information

Out-of-date knowledge poses a significant limitation for pre-trained LLMs. To answer a question, these models rely heavily on the data used during their training.

This can affect the model's performance in tasks that require current information. Therefore, keeping the model updated with the latest data is essential to mitigate this limitation. However, updating the parameters of the model is expensive and can lead to unexpected side effects [18].

Hallucinations

Hallucination occurs when a language model generates responses that are inaccurate or nonsensical. These responses may seem plausible but are based on erroneous information. These hallucinations can be intrinsic (based on the input prompt) or extrinsic (partially or not at all based on the input prompt) [18].

Overcoming the Limitations

Hallucinations can be reduced through augmentation. By leveraging an external knowledge base, the likelihood of generating hallucinated responses can be lowered. Shuster et al. [19] achieved state-of-the-art performance on knowledge-intensive tasks by employing retrieval augmentation, and their best model significantly reduced hallucinations. Lewis et al. [6] discuss Index hot-swapping: *"An advantage of non-parametric memory models like RAG is that knowledge can be easily updated at test time."*. This effectively addresses the challenge posed by outdated information.

2.3 Evaluation

2.3.1 Recall

Recall stands out as one of the most important metric for retrieval evaluation. A recall-oriented task is one where the objective is to maximize the number of relevant items retrieved from a huge pool of data. It measures the proportion of relevant documents retrieved by the system among all documents available in the knowledge

base. Recall score is number between 0 and 1, a higher score indicates that the system captures a significant portion of relevant information based on the search query. The recall score is proportional to the number of items retrieved. A recall score of 100% would be achieved by a system that retrieves all the documents.

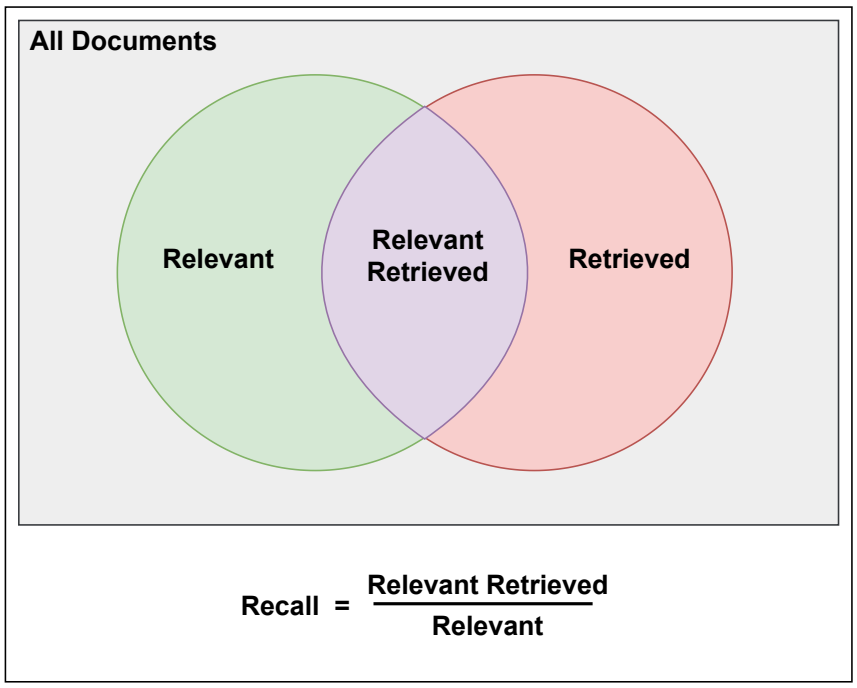

Figure 2.4: *Recall*

While precision (the proportion of retrieved items that are relevant) is also important, the choice of the evaluation metric depends on the specific use case. In a question answering system that relies on factual information, prioritizing recall is more logical. M. A. C. Soares and F. S. Parreiras [20] highlights the importance of recall, *"Most of fact QA systems should use Recall as a measure metric since it does not matter how high the false positive rates are, if there are high true positive rates, the result will be good."* Recall, when considered independently, is an unranked measure of performance as it does not take into account the order/ranking of retrieved items. A variation of recall, known as recall@n, can be used as a ranked measure of performance. It measures the proportion of relevant items among the top n retrieved

items, where n is the predefined cutoff point.

$$Recall@n = \frac{RelevantRetrieved@n}{Relevant} \tag{2.7}$$

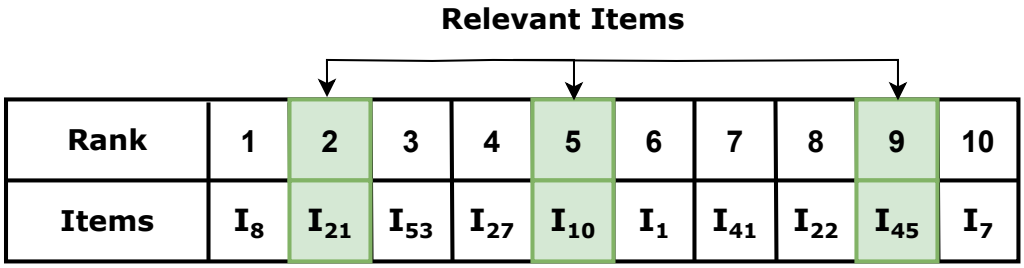

Figure 2.5: *Recall@n: R@3 = 0.33, R@5 = 0.66, R@10 = 1.0*

2.3.2 Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank (MRR) is a ranking-based metric. It evaluates the performance by considering the rank of the first relevant document retrieved. MRR provides a focused assessment of how quickly the correct document can be found in the retrieved ranked list of results. This method is sensitive to the rank position and considers the rank of the top relevant document in the retrieved list. It is calculated as the mean of the reciprocal of the rank of the first relevant document of each query:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{2.8}$$

Where $|Q|$ is the total number of queries and $rank_i$ represents the rank of the first relevant document for the query i.

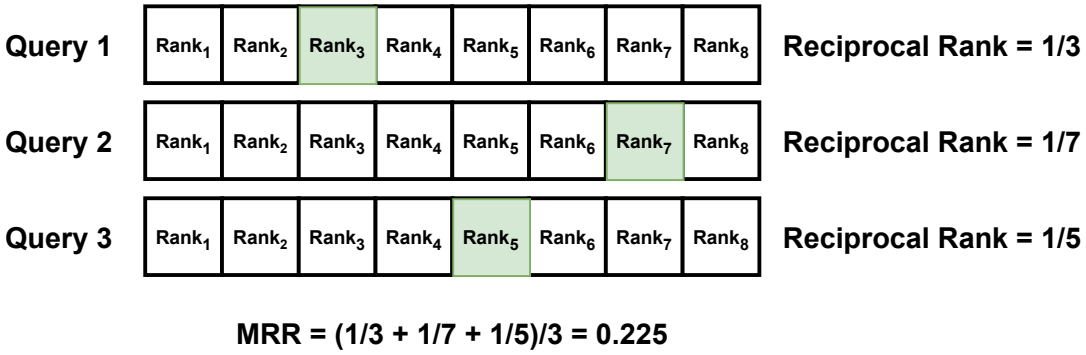

Figure 2.6: *Example calculation of MRR*

MRR score is a value between 0 and 1. A higher value of MRR indicates that the relevant results are found at the top of the ranked list of results, meaning a more efficient retrieval process.

2.3.3 Evaluation Datasets

This section describes some of the famous benchmarking datasets that provide a standard measure for comparing the performance of different RAG-based systems.

Natural Questions (NQ) dataset is a huge dataset created by Google. It consists of real queries submitted to the Google search engine along with their corresponding relevant documents and annotated long and short answers. *"The public release consists of 307,373 training examples with single annotations."* [21].

Example
Question: what color was john wilkes booth's hair
Wikipedia Page: John_Wilkes_Booth
Long Answer: Some critics called Booth "the handsomest man in America" and a "natural genius", and noted his having an "astonishing memory"; others were mixed in their estimation of his acting. He stood 5 feet 8 inches (1.73 m) tall, had jet-black hair, and was lean and athletic. Noted Civil War reporter George Alfred Townsend described him as a "muscular, perfect man" with "curling hair, like a Corinthian capital".
Short Answer: jet-black

Figure 2.7: *This figure has been adapted from the paper "Natural Questions: a Bench- mark for Question Answering Research" [21]*

TriviaQA, another famous dataset for natural language comprehension and question-answering. It contains around 650,000 complex questions and evidence documents from Wikipedia [22].

Example
Question: The Dodecanese Campaign of WWII that was an attempt by the Allied forces to capture islands in the Aegean Sea was the inspiration for which acclaimed 1961 commando film?
Answer: The Guns of Navarone
Excerpt: The Dodecanese Campaign of World War II was an attempt by Allied forces to capture the Italian-held Dodecanese islands in the Aegean Sea following the surrender of Italy in September 1943, and use them as bases against the German-controlled Balkans. The failed campaign, and in particular the Battle of Leros, inspired the 1957 novel The Guns of Navarone and the successful 1961 movie of the same name.

Figure 2.8: *This figure has been adapted from the paper "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension" [22]*

Cohesive Long-form Answers from Passages in Natural Questions (CLAPNQ) This dataset is relatively newer and specifically designed to evaluate RAG-based systems. It was curated using the NQ dataset and provides long-form

answers by integrating multiple passages from the NQ dataset to form a single coherent response. While this dataset can be used to evaluate all parts of a RAG system, including retriever, generator and the entire RAG pipeline, it is particularly useful for assessing the quality of detailed and cohesive response generation [23].

Example
Question: what is the story of call of duty zombie
Title: Call of Duty: Black Ops III
Reference Answer: Call of duty: Black Ops III takes place in 2065 in a world facing upheaval from climate change and new technologies. The game features a standalone Zombies mode, and a " Nightmares " mode which replaces all enemies as zombies.
Passage: Black Ops III takes place in 2065, 40 years after the events of Black Ops II, in a world facing upheaval from climate change and new technologies. Similar to its predecessors, the story follows a group of black ops soldiers. The game's campaign is designed to support 4-player cooperative gameplay, allowing for bigger, more open level design and less corridor shooting. As the player character is cybernetically enhanced, players have access to various special activities. The game also features a standalone Zombies mode, and a "Nightmares" mode which replaces all enemies as zombies.

Figure 2.9: *This figure has been adapted from the paper "CLAPNQ: Cohesive Long-form Answers from Passages in Natural Questions for RAG systems" [23]*

Machine Reading Comprehension Dataset (MS MACRO), created by Microsoft, is another widely used dataset for benchmarking retrieval tasks such as document retrieval, passage retrieval, key phrase extraction and question-answering. It consists of 1,010,916 real-word queries received by Microsoft through Bing with human-generated answers along with 8,841,823 passages for answer generation [24].

Example
Question: what is a corporation?
Answer: A corporation is a company or group of people authorized to act as a single entity and recognized as such in law.
URL: http://www.wisegeek.com/what-is-a-corporation.htm
Passge: A company is incorporated in a specific nation, often within the bounds of a smaller subset of that nation, such as a state or province. The corporation is then governed by the laws of incorporation in that state. A corporation may issue stock, either private or public, or may be classified as a non-stock corporation. If stock is issued, the corporation will usually be governed by its shareholders, either directly or indirectly.

Figure 2.10: *This example has been utilized from the MS MACRO dataset [24]*

Benchmarking IR (BEIR) provides a comprehensive evaluation benchmark for retrieval tasks that compares different types of information retrieval architectures. BEIR evaluation is based on 18 publicly available datasets covering diverse tasks. NQ and MS MACRO datasets are also a part of BEIR evaluation, and this protocol is specifically designed to evaluate information retrieval models [25].

2.4 Prompt Engineering

Prompt engineering has emerged as an important aspect of natural language processing. As LLM technologies continue to evolve, prompt engineering has become increasingly popular as a new discipline of scientific research. White et al. [26] defines prompts as *"instructions given to an LLM to enforce rules, automate processes, and ensure specific qualities (and quantities) of generated output."* The role of well-crafted prompts is crucial in harnessing the full potential of these powerful language models. Prompt engineering on the other hand, refers to the design and optimization of these prompts to enhance the performance of the LLM for specific tasks [27]. Prompt engineering contains various techniques such as prompt design and tuning, prompt evaluation, multimodal prompting, zero-shot and few-shot prompting.

L. Giray [28] identifies four primary elements for defining an effective prompt:

1. ***Instruction*:** Clear instructions provide guidance and define the scope and nature of the task. They help avoid ambiguity and misinterpretation, leading to more precise and relevant outputs.

2. ***Context*:** Context provides background information that enables the model to adapt its response accordingly. It enhances the model's comprehension and improves the relevance of outputs.

3. ***Input data*:** Input data represents the actual information that the user wants the model to process. It can include text, images, documents, etc.

4. ***Output Indicator*:** Output indicators specify the desired format or structure of the model's output. They allow users to control the presentation and organization of the model's outputs, ensuring consistency responses.

S. Ekin [27] emphasizes that one of the most effective approaches to guide the model towards the desired output is through an iterative cycle of testing and refining prompts. Another crucial element of prompt engineering involves finding the right balance between the model's creativity at generating responses and the specific task requirements. For instance, allowing the model to be creative is advantageous in contexts where innovation, exploration, or user engagement are desired outcomes, such as creative writing, artistic expression, entertainment, etc. On the other hand, restricting model's creativity is necessary in situations where precision and accuracy are critical for achieving the intended objectives, tasks such as medical diagnosis and treatment, financial and investment advice, and factual information retrieval necessitate the generated response to be deterministic.

2.4.1 Prompting Techniques

Zero-Shot Prompting

In zero-shot prompting, the language model is instructed to generate a response without including any specific examples related to the given task. In this scenario, the model is expected to generate a response based exclusively on its pre-existing knowledge and understanding of the language.

One-Shot Prompting

In one-shot prompting, the language model is provided with a single example or demonstration related to the given task. This single example guides the model's response generation process to produce the expected outputs.

Few-Shot Prompting

Few-shot prompting extends the concept of one-shot prompting, where the model is provided with multiple examples related to the task instead of single specific example. These examples provide additional context and supervision, allowing the model to generalize and adapt its behavior more effectively.

The study by Brown et al. [29] compares zero-shot, one-shot and few-shot settings with state-of-the-art fine-tuned language models on 9 different natural language tasks. The results indicate that for the majority of tasks, as the parameter size increases, the performance achieved in the few-shot setting is comparable, and in some cases, even superior to that of state-of-the-art fine-tuned language models.

2.5 Research Questions/Objectives

The goal of this research is to identify opportunities for improvement in the current system by addressing the following research questions:

1. Does the keyword generation/extraction process impacts the performance of the system, or can it be omitted?

2. How does different prompting techniques affects the performance of the system?

3. Exploring additional parameters that could impact the system's performance

3 Dataset Creation

3.1 Data Collection

The data used in this study consists of the documentation data and the user queries. The documentation data is openly available at the company's website. It contains information regarding various software solutions offered in multiple format. This data were scraped from the company's website and after processing, it was indexed in Amazon OpenSearch Service [30], formerly known as Amazon Elasticsearch Service. The indexed documents consist of the following content type: Articles, Bulletins, Courses, Product Guides, Release Notes and Video Tutorials. In total, 55,805 pages are indexed in Amazon OpenSearch Service in English language. A well-constructed dataset is crucial for effective evaluation. This study requires a dataset that establishes the ground truth, represents the real-world user queries, facilitates the calculation of performance metrics and provides a comparative analysis. The dataset used in the study is primarily prepared using the user queries received through emails to the customer support. It contains 150 user queries and has the following attributes:

Dataset	
Attribute	**Description**
Product	The name of the software product.
Category	The category related to the user query.
Version	The version of the Product, when the user query is related to a specific version.
Original_Question	The original question asked by the user.
Simplified_Question	The simplified form of the original user question.
Resolution_Link	The Resolution URL to the company's website containing the possible solution to the question.

Table: Attributes with their descriptions in the User Query Dataset.

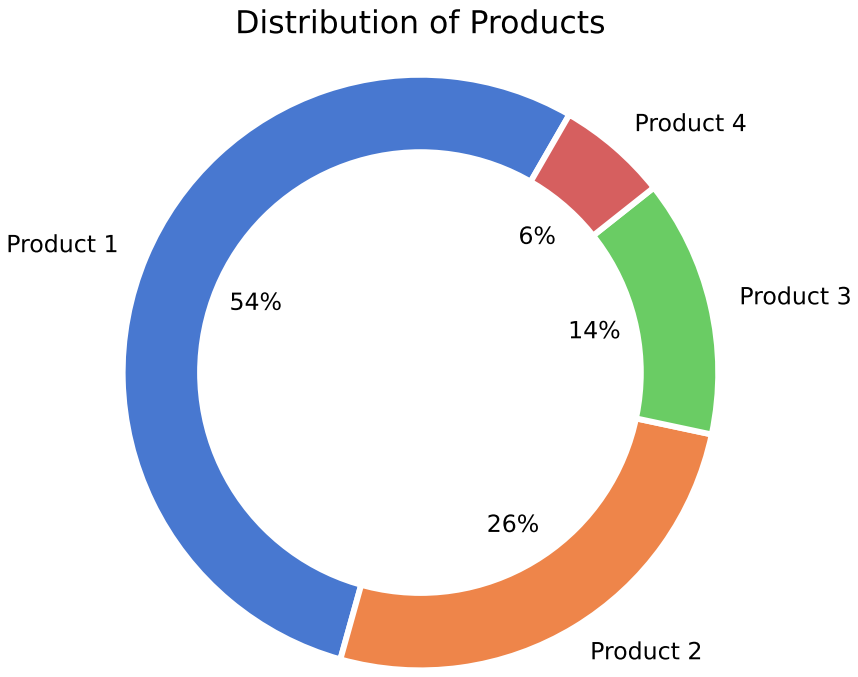

Figure 3.1: *The chart represents the relative proportions of each product in the overall distribution. Product 1 dominates with over half of the distribution.*

The evaluation dataset has been filtered to have only those queries that have a valid **Resolution_Link** sent to the user as a resolution for the issue they are facing. In some cases, it might happen that the solution cannot be found in the documentation and it requires human intervention to address the issue.

4 Methodology

4.1 Retrieval Methodology

4.1.1 Keyword Generation

The first step is to generate or extract keywords from the simplified and original user questions represented by **Simplified_Question** and **Original_Question** attributes respectively. These keywords are generated using the OpenAI model [31] with a predefined prompt, referred to as the default prompt. The model specifications are as follows:

Model name: gpt-4-32k

Model version: 0314

Context window: 32,768 Tokens

Capacity: 30,000 Tokens Per Minute

Training data: Up to Sep 2021

The default system and user prompts used to instruct the LLM are as follows:

Default System Role Prompt

"You are an AI assistant that helps generate keyword searches."

Default User Prompt

"Generate a keyword search from the following statement or question. Do not include commas in your response:{question_placeholder}"

Original question and the generated keywords	
Question	In Column "H-Description" we added plate thickness with formula. We don not want inch symbol for the plate thickness. E.g. Description of the plate should read as PL3/4 not PL3/4". Please provide solution for the same.
Keywords	Column H-Description plate thickness formula remove inch symbol PL3/4

Table: An example of keywords generated using Original Question.

Simplified question and the generated keywords	
Question	How to remove inch symbol in profile names?
Keywords	remove inch symbol profile names

Table: An example of keywords generated using Simplified Question.

The generated keywords serve as the basis for document retrieval. This results in addition of two new attributes to the dataset, namely **Keywords_SQ_default_prompt** and **Keywords_OQ_default_prompt**. These represent keywords generated from simplified and original questions, respectively, using the default prompt.

4.1.2 Document Retrieval

Amazon OpenSearch Service is a distributed search and analytics engine popularly used for indexing and searching large volumes of unstructured data. The generated

keywords are used to execute search queries to Amazon OpenSearch Service. For evaluation, the top 20 matching documents are retrieved based on their relevance score to the input keywords. These documents are returned from the search service in descending order of their relevance scores where the 1^{st} retrieved document has the highest matched and the 20^{th} document has the lowest match with the search keywords. In addition to the keywords, the search query contains three additional parameters: Product, Version and Environment. These parameters further refine and narrow down the search for increased specificity. The query must contain a valid product name. However, The version and environment parameters can either be set as "Not version-specific" and "Not environment-specific", or they can specify a particular version and environment for a specific product. For each retrieved document, the document URL is extracted, which contains a reference to the source of information. Finally, the results of each query run are stored as a list of 20 URLS. The queries are executed for four cases and saved under four new attributes of the evaluation dataset as follows:

1. **Results_SQ_default_prompt**: Results of the query executed on simplified questions using the default prompt for 'Not version-specific' case.

2. **Results_SQ_VS_default_prompt**: Results of the query executed on simplified questions using the default prompt for 'version-specific' case.

3. **Results_OQ_default_prompt**: Results of the query executed on original questions using the default prompt for 'Not version-specific' case.

4. **Results_OQ_VS_default_prompt**: Results of the query executed on original questions using the default prompt for 'version-specific' case.

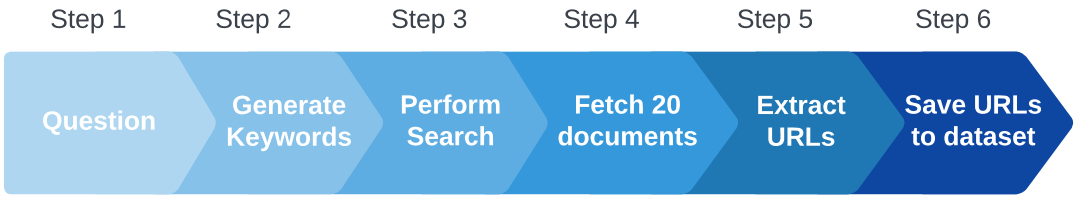

Figure 4.1: *The retrieval process*

4.2 Evaluation Methodology

The experiments conducted in this study utilize two evaluation metrics for measuring the performance of retriever component: Mean Reciprocal Rank (MRR) and Recall@n. The evaluation compares performance for simplified and original questions in both version-specific and non-version-specific scenarios. The newer versions of the software may introduce new and additional features, while also retaining all the features present in the older versions. This results in redundancy or overlapping of information in the documentation across different versions. Specific modifications have been made to the evaluation metrics used in the experiments to better align with the objective of the study and characteristics of the evaluation dataset.

Figure 4.2: *An example of relevant and retrieved set.*

Figure 4.2 represents an example of the relevant and retrieved set of URLs when version is not specified in the search query. The software version is represented by the year in the URL. The retrieved URL highlighted in green (at rank 7) represents an exact match of the relevant URL. However, it is important to note that the retrieved URLs highlighted in orange (rank 4 to 8) belong to the different versions of the software containing identical information to the retrieved URL at rank 7. Therefore, calculating the evaluation metrics using the exact match of the relevant

URL in the retrieved set leads to inaccurate and erroneous calculations, particularly for the 'Not version-specific' case. In order to address this issue, the evaluation metrics have been adjusted to consider the first URL containing identical information instead of relying solely on the exact match of the relevant URL. In other words, the retrieved URL at rank 4 will be considered for calculation of the metrics instead of the URL at rank 7, since the URL at rank 4 contains identical information to the URL as rank 7.

4.2.1 Preliminary Evaluation

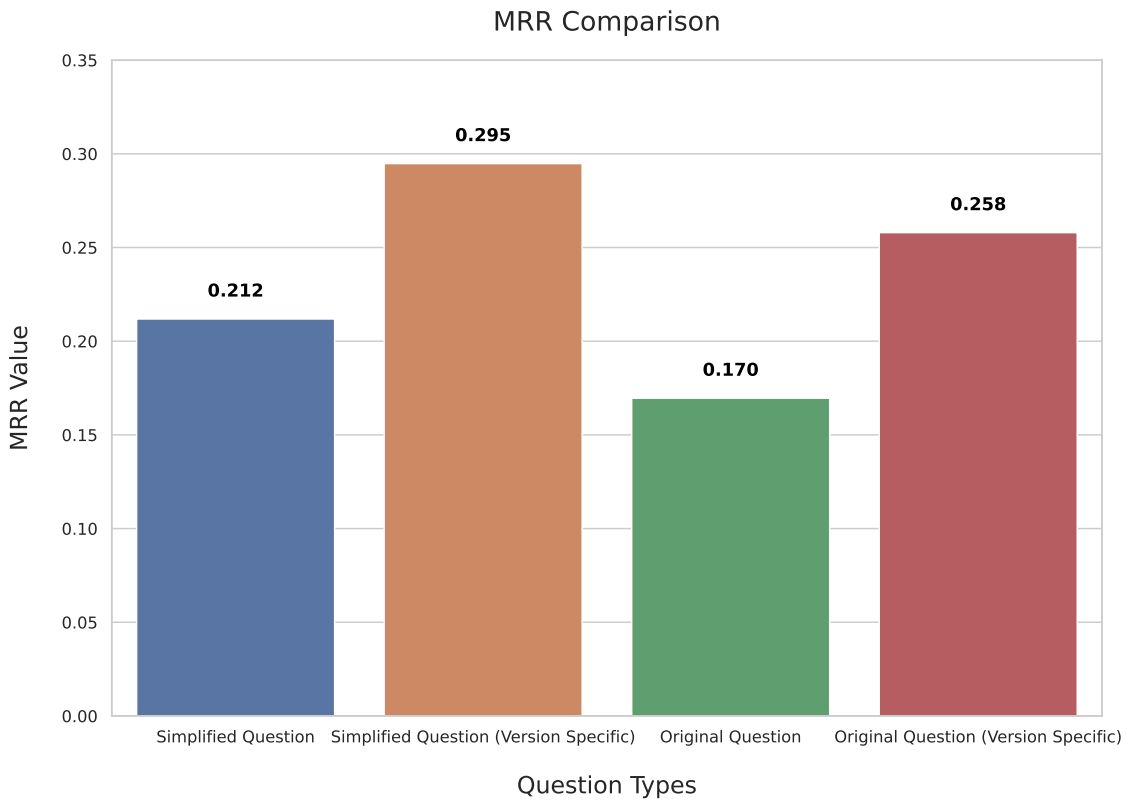

Figure 4.3: *Illustrates the Mean Reciprocal Rank for all 4 cases*

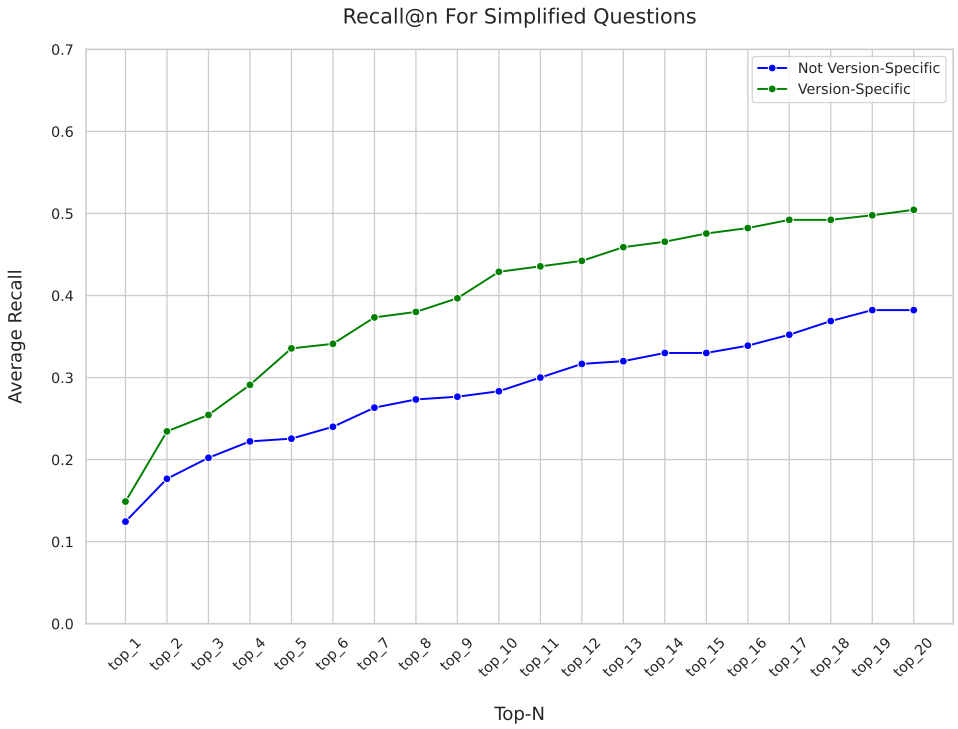

Figure 4.4: *Recall@n for simplified questions*

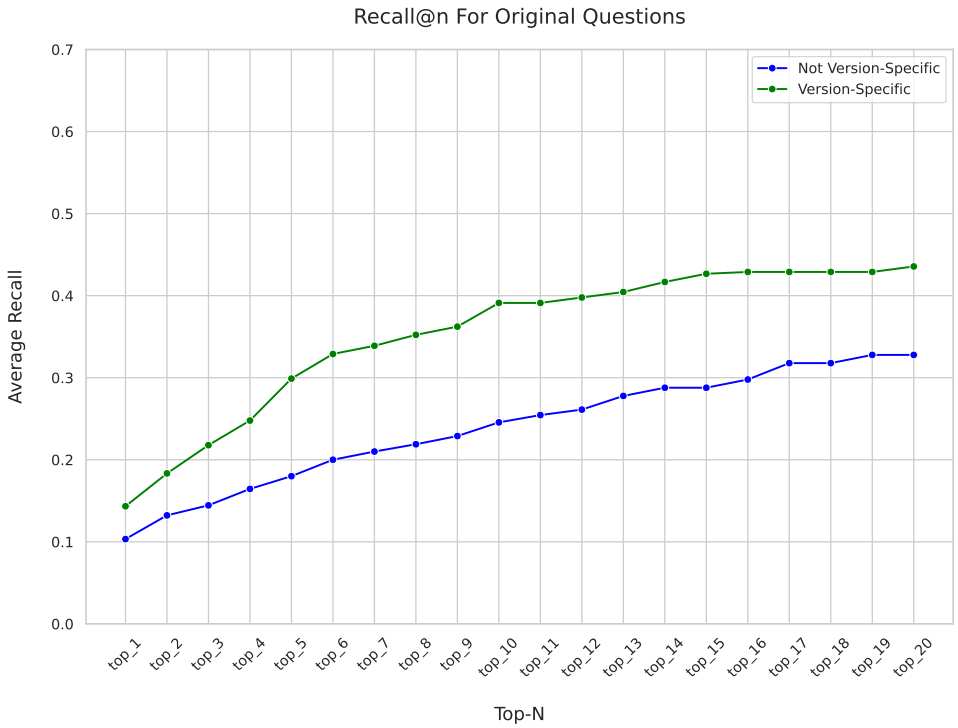

Figure 4.5: *Recall@n for original questions*

Preliminary results show that specifying the version in the query has a significant impact on both of the evaluation metrics. It can also be noted that performance with the original question types remains lower than the simplified questions.

4.2.2 Error Analysis

The preliminary evaluation of the retrieval augmentation system shows suboptimal performance. When manually testing the generated answers by the system, inconsistencies were observed between the retrieved documents and the expected ground truth labels. Error analysis is conducted to identify and address inconsistent ground truth labels in the evaluation dataset. The aim of error analysis is to rectify the missing labels and improve the quality of the dataset. A subset of questions is identified during error analysis, the questions were filtered from the dataset for which the recall for the top 20 retrieved documents was zero. This subset of questions was manually tested with the system and show that the system still produces correct answers despite the recall being zero. Upon closer examination, 56 questions were identified as incorrectly labeled and lacked associated resolution URLs in the evaluation dataset. While the majority of answers can be found within one single URL, there are instances where a significant portion of the answer can be located at one specific URL and other smaller parts can be located across other URLs. After reviewing, the dataset is updated with correct labels for questions that were inaccurately labeled or missing resolution URL. The results after error analysis are plotted below and works as a benchmark of the current system performance and is used for testing the system with different settings.

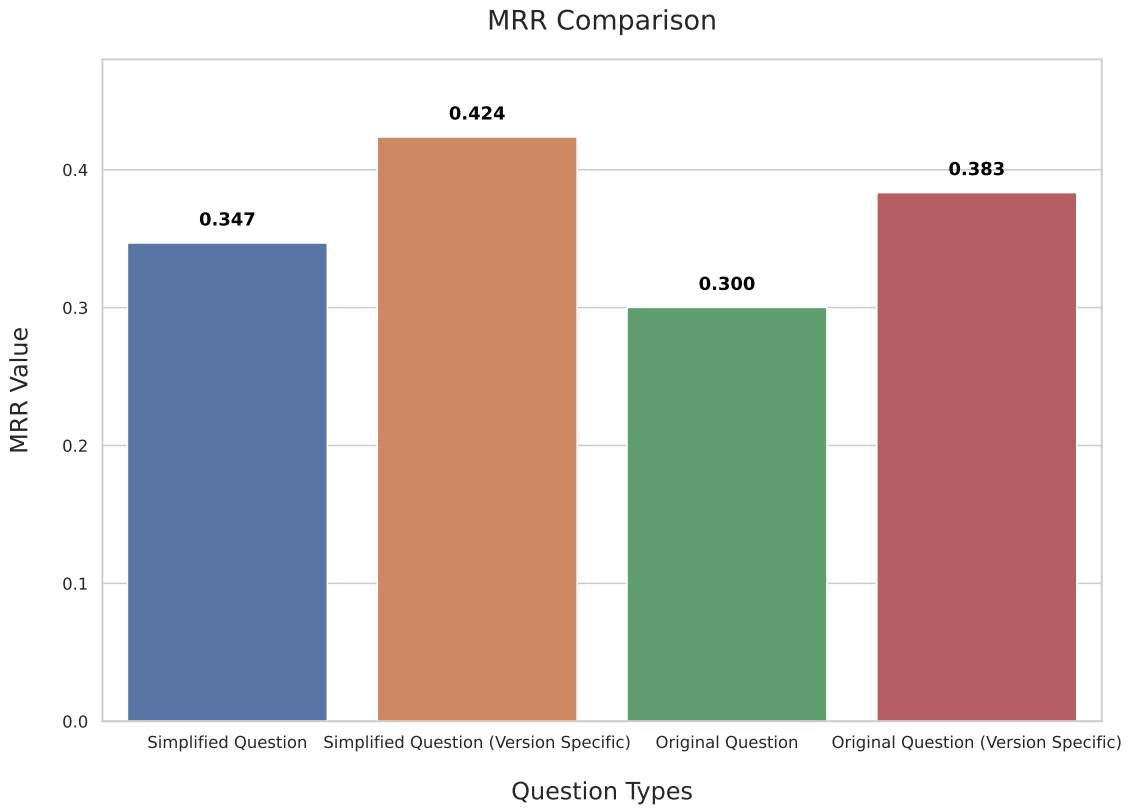

Figure 4.6: *The Mean Reciprocal Rank after error analysis*

It can be observed that the MRR for non-version-specific case has increased significantly for both simplified and original questions after performing the rectification. However, the system still performs better when supplemented with the version within the search query. In the case of simplified questions, an increase of 22.19% can be noticed. While in the case of original question, this increase is slightly higher, standing at 27.66%. These MRR scores indicate the average position/rank at which the first relevant document can be found in the retrieved results. The higher the average rank, the better the results. In simplified questions, we get the first relevant document, on average, at rank 2.88 and 2.35 for non-version-specific and version-specific case respectively. Similarly, for original question, the average ranks are 3.33

(non-version-specific case) and 2.61 (version specific). However, it is important to note that the first relevant hit might not contain all the information required to produce an accurate answer.

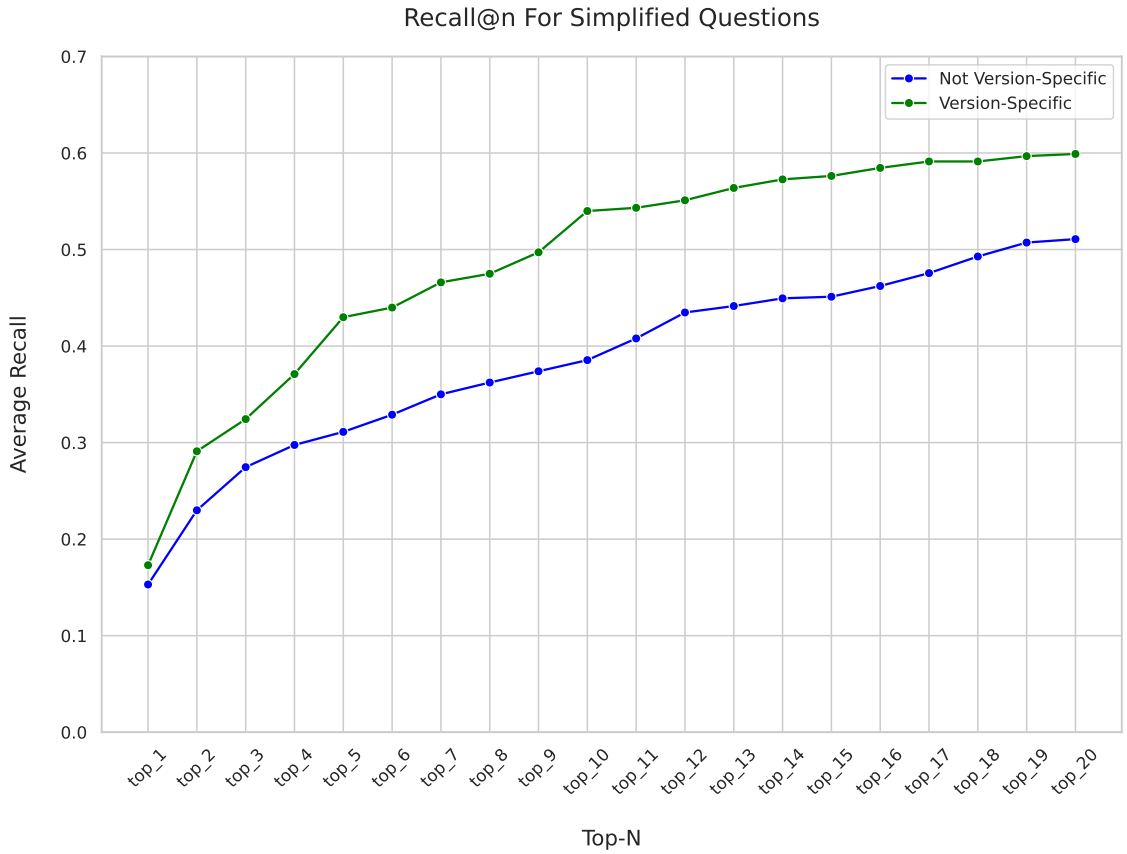

Figure 4.7: *Recall@n for simplified questions after error analysis*

In the case of simplified questions, the recall values for version-specific cases are consistently higher than the recall for non-version-specific case across all values of n. Recalls for both cases generally increases as the number of documents retrieved increases. However, it can be observed that the rate of increase in recall for version-specific case is slightly steeper compared to non-version-specific, especially before n

= 10. This indicates that the version specific case benefits more with an increasing number of top-n. The gap between version-specific and non-version-specific cases widens as n increases. Although, after n crosses 15, the recall for both cases appear to converge slightly.

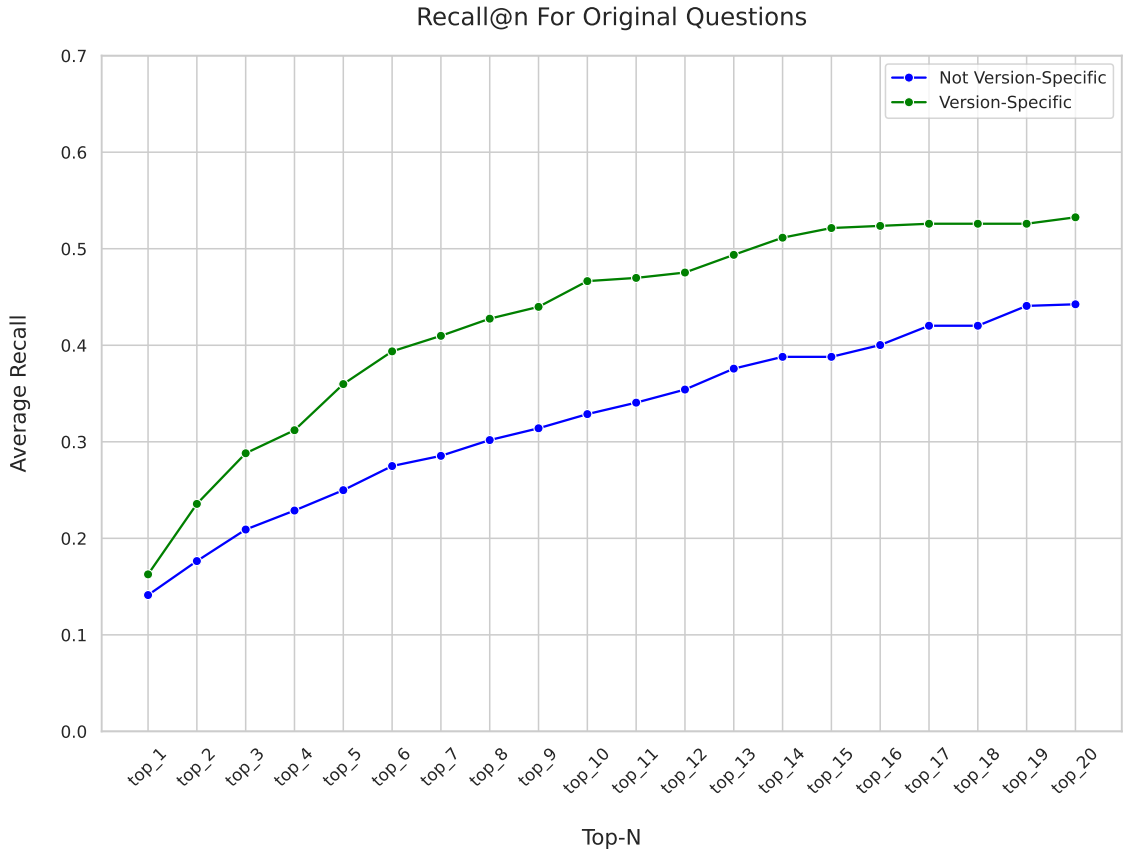

Figure 4.8: *Recall@n for original questions after error analysis*

Results from original questions show a similar trend to the simplified questions. Recall in version-specific consistently stays over non-version-specific values. It can be seen that in the version-specific case, the recall stays plateaued beyond n = 15 and barely increases. On the other hand, for the non-version-specific the recall consistently increases till n = 14, after which it still increases but with minor fluctuations and doesn't seem to saturate.

5 Results

5.1 Evaluation Without Keyword Generation

This section addresses the impact of keyword generation on the retrieval performance of the system. In general, the keyword generation enhances the performance of the document retrieval process. However, it is crucial to assess whether incorporating this step has a significant effect on the retrieval process, since the addition of this step leads to additional requests, which is expensive and causes more delay in the answer generation. In order to evaluate performance in this case, the keyword generation process, as shown in Step 2 in Figure 4.1, is completely bypassed, and the entire user question is used for performing the search and retrieving documents. This experiment serves as a baseline to compare against the retrieval results achieved by including the keyword generation step.

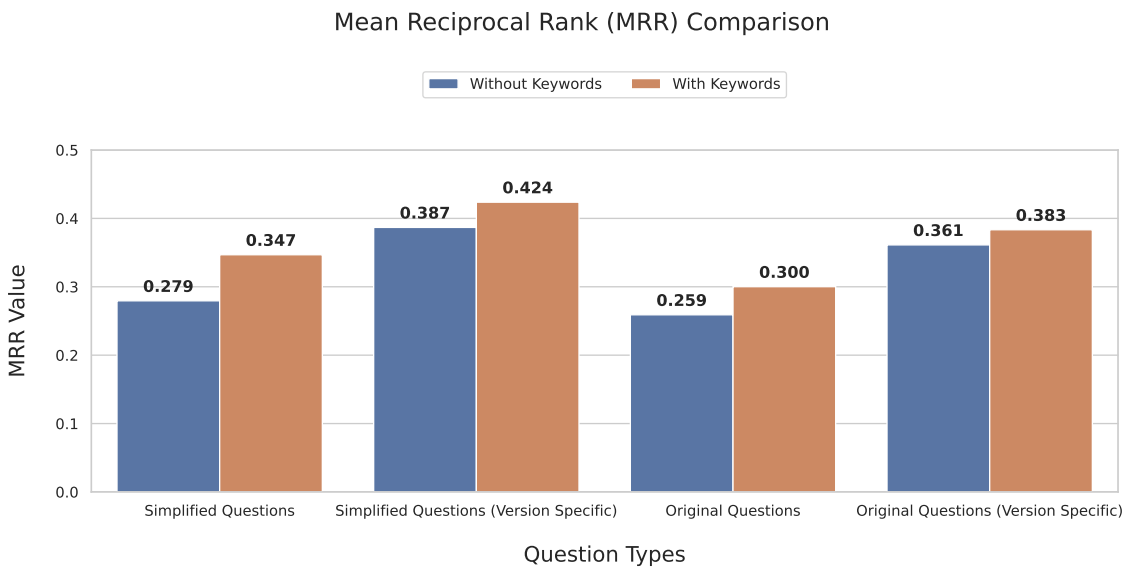

Figure 5.1: *The Mean Reciprocal Rank with and without keyword generation*

The comparison of MRR shows that keyword generation has a positive effect on the ranking of the relevant retrieved documents. It can be seen that the difference is much higher for non-version specific case for both simplified and original questions, with MRR increasing by about 24.37% and 15.83% respectively. On the other hand, in the case of the version-specific scenario, the increase is less than 10% for both types of questions. Furthermore, keyword generation leads to a higher increase in the MRR results for simplified questions as compared to original question type. This indicates that simplified questions are likely to contain more concise and focused keywords that directly match with the terms in the documents. On the other hand, original questions are more descriptive in nature and contain complex language which makes it difficult for the system to extract precise keywords.

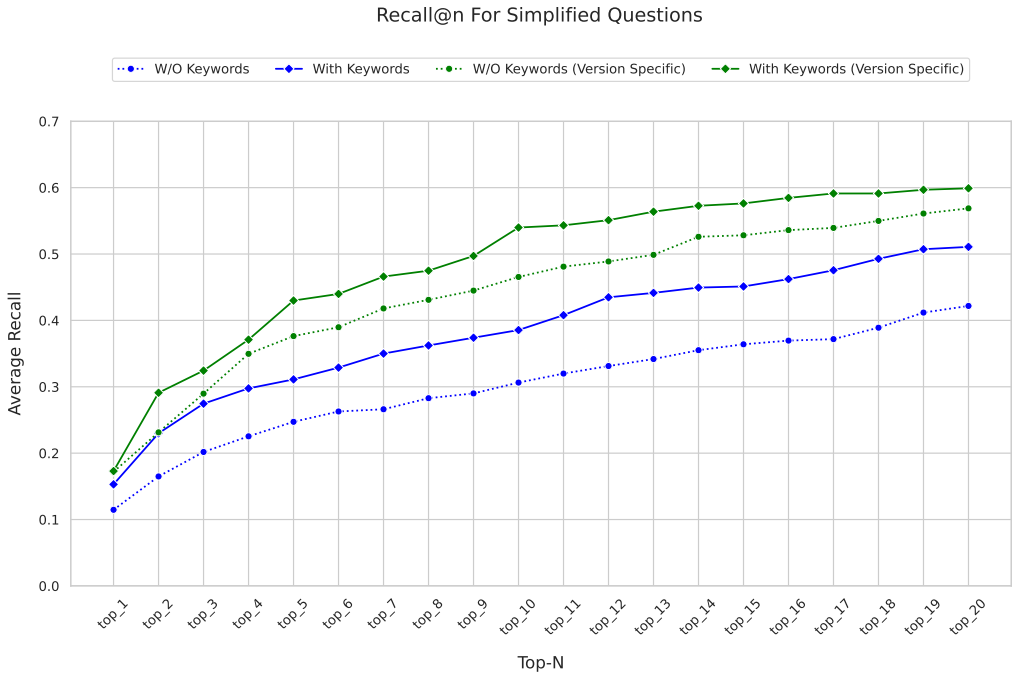

Figure 5.2: *Recall@n for simplified questions with and without keyword generation*

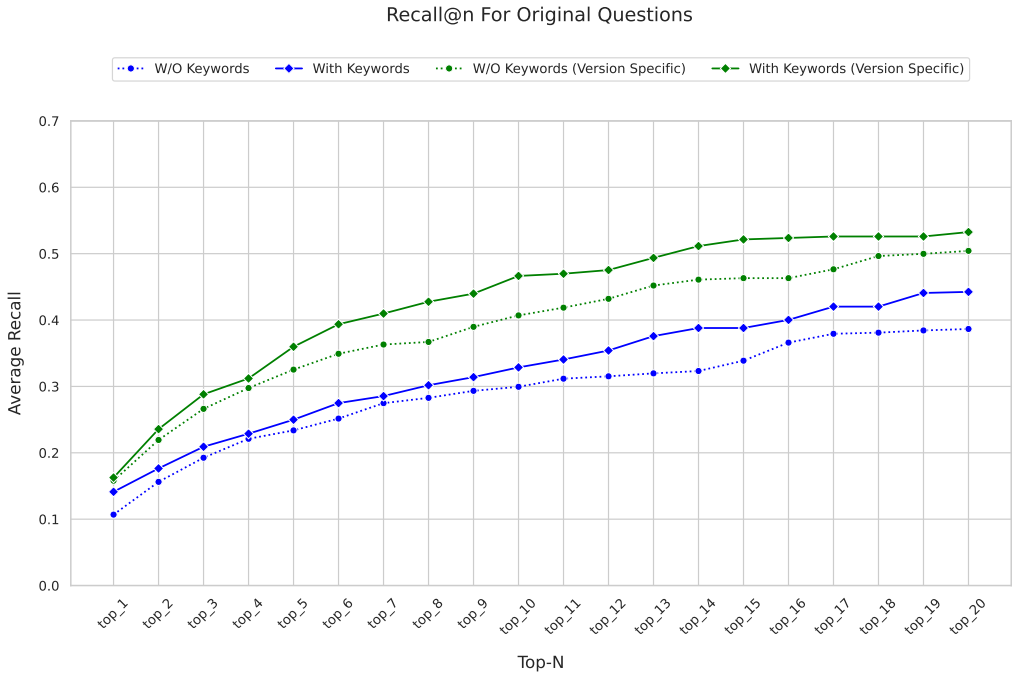

Figure 5.3: *Recall@n for original questions with and without keyword generation*

Similar to MRR, the recall results also show a positive impact of keyword generation on both question types in both version-specific and non-version-specific scenarios. For simplified questions, the recall results with keyword generation consistently remain above the results without keyword generation from the beginning. A similar trend can be observed in original questions, the lines closely follow each other up to top 5 for the version-specific case and up to top 10 for non-version-specific case, after which the difference increases in both the cases. This suggests that for lower values of top-n, retrieval performance is similar regardless of keyword extraction. As we move down the list of retrieved documents, keyword generation becomes more impactful in identifying relevant documents that are ranked lower in the retrieval list.

The overall comparison shows that generating/extracting relevant keywords from the questions improves both the MRR and recall performance of the system. It enhances the system's ability to retrieve a greater number of relevant documents for every value of n. The ranked position of retrieved documents improves as well.

5.2 Prompt Evaluation

Prompt engineering plays an important role in generating the desired outputs from language models. This study experimented with five different prompts to generate keywords for retrieval. The aim is to examine how different variations of prompts affect the relevance and specificity of extracted keywords. The prompts used are as follows:

5.2.1 Prompt #1

System Prompt

"Your role is to assist users in refining their search queries for software products. Generate keywords that will yield more accurate and relevant results from the search engine."

User Prompt

"Improve the search query by generating keywords from the provided statement or question. Please avoid including commas in your response: {question_placeholder}"

Comparing with the default prompt, Prompt #1 contains additional task specific information. It tells the LLM that the keywords will be used to perform a search and the LLM is expected to produce keywords that leads to more accurate search results.

5.2.2 Prompt #2

System Prompt

"Given a user query related to [Company] products ('Product 1', 'Product 2', 'Product 3', 'Product 4'), generate a list of keywords that can be used to effectively search Amazon OpenSearch service (Elasticsearch) for relevant information. The keywords should be specific to the user's query, the functionalities of the software product(s) mentioned, and potentially include synonyms or related terms to capture a broader range of relevant information."

User Prompt

"Generate keywords (separated by white space) relevant to {product_placeholder} for the following user query: {question_placeholder}"

The system prompt #2 contains information about the company and mentions all the products offered by the company. Additionally, The user prompt #2 also contains the product name in the query.

5.2.3 Prompt #3

System Prompt

"You are an AI assistant that helps generate keyword searches for a chatbot supporting [Company] construction software products, including Product 1, Product 2, Product 3, and Product 4. The keywords should be concise, relevant, and suitable for searching technical documentation, user guides, FAQs, and support articles related to these products."

User Prompt

"Given the following user statement or question: {question_placeholder}

Generate one set of relevant keywords that can be used for searching [Company's] documentation and support resources. Include specific product names, features, or components in the keywords where applicable. Here is an example:

User query: 'Hi, Dia symbol not showing in template showing only block, how to add this in template?'
Keywords: add diameter symbol Template Editor

User query: 'How can I create a bolt in the catalog?'
Keywords: adding bolts managing bolt assemblies importing bolt assemblies

Only generate keywords"

Prompt #3 is an example of few-shot prompting and is an extension of Prompt #2. The user prompt contains some example queries from the dataset along with the expected keywords.

5.2.4 Prompt #4

System Prompt

"You are an AI assistant that helps generate keyword searches for {product_placeholder}"

User Prompt

"Given the following user statement or question: {question_placeholder}
Generate one set of relevant keywords separated by white space. Include specific features, or components in the keywords where applicable."

Prompt #4 only contains the specific product related to the query, unlike the previous prompts that contain all the products.

5.2.5 Prompt #5

System Prompt

"You are an AI assistant that helps generate keyword searches for {product_placeholder}. Generate relevant keywords that can be used for searching [Company's] documentation and support resources. Here are some example:

Example 1:
Question: Why does the frame view appear as straight line?
Keywords: frame view straight line issue display [Product]

Example 2:
Question: Why do I get error 'the job # does not match the current job' when trying to import an XML file in [Product]?
Keywords: [Product] XML file import error job number mismatch current job

Example 3:
Question: Download and installation of Smart3d Interoperability tool.
Keywords: download installation Smart3D Interoperability tool [Product]

Example 4:
Question: Would you be able to advise on how easy it would be to convert from a [Product] 3D model to a Revit 3D file?
Keywords: convert [Product] 3D model Revit 3D file

Example 5:
Questions: I am having trouble performing the connection designs. I don't see the option to export the connections to the [Product] connection design software?

Keywords: connection designs export connections [Product] connection design software"

User Prompt

"Given the following user statement or question: {question_placeholder}

Generate relevant keywords separated by white space."

Prompt #5 is another form of few-shot prompting. This prompt contains more examples, and these examples are carefully filtered from the dataset. It contains only those questions for which the recall is 1 with all the previous prompts. Similarly, these examples contain keywords that were common across all the previous prompts. Therefore, these examples represent the perfect combinations of questions and corresponding keywords.

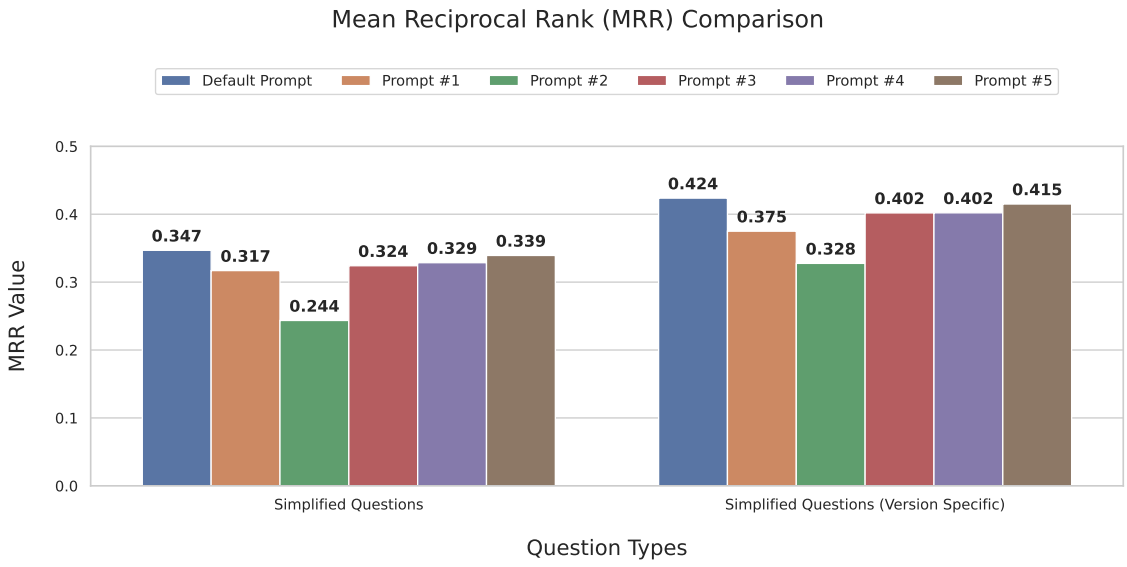

Figure 5.4: *The Mean Reciprocal Rank for simplified questions with different prompts*

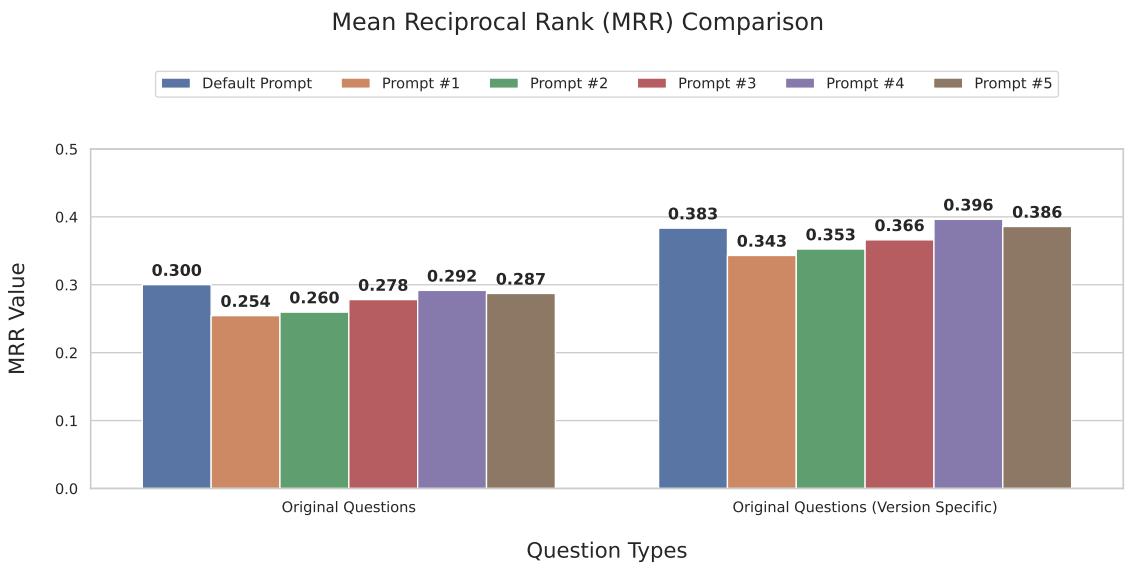

Figure 5.5: *The Mean Reciprocal Rank for original questions with different prompts*

Figure 5.4 and Figure 5.5 indicate that the default prompt achieves the highest MRR in all the cases except for the Original Questions (Version Specific) case, where Prompt #4 outperforms the default prompt by a small margin. Prompt #1 and prompt #2 underperform in all four cases. However, Prompt #2 achieves a higher MRR with original questions compared to simplified questions. Version-specific scores are consistently higher than the non-version-specific cases for both question types across all prompts. For original questions, the scores closely follow each other across all the prompts, whereas for simplified questions, the scores are more deviated.

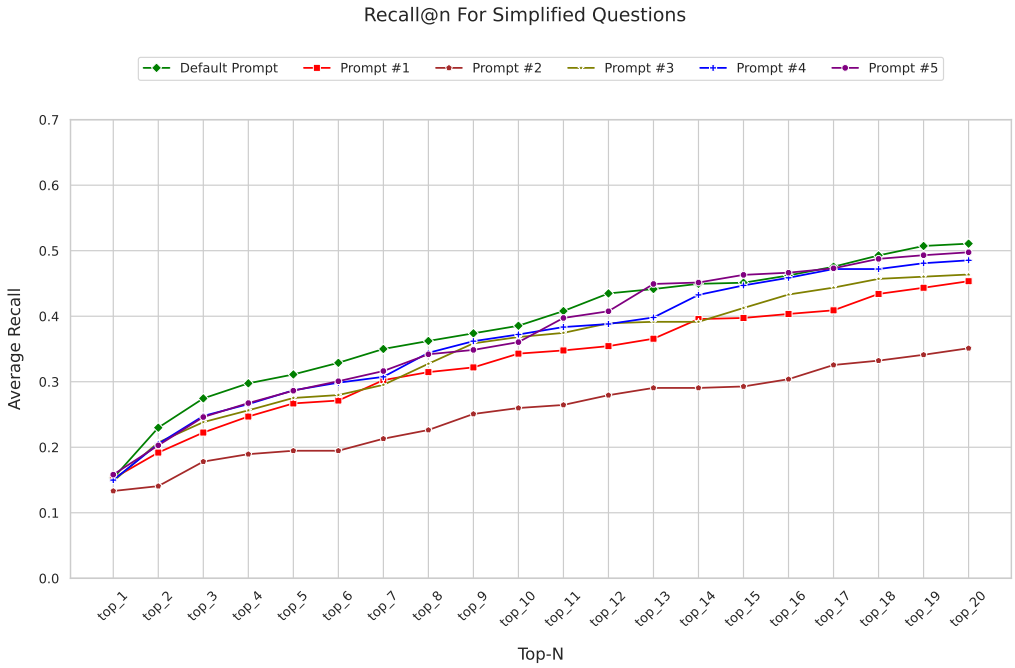

Figure 5.6: *Recall@n for simplified questions using different prompts*

For simplified questions non-version-specific case (Fig. 5.6), Prompt #2 starts with the lowest recall and remains lowest throughout all the values of n. Recall with default prompt constantly stays higher than other prompts for initial values of n, up to top-8, after which the difference between default prompt and prompt #4 and #5 decreases. Figure 5.6 shows a near-linear trend, where scores across all the prompts increases steadily as the number of top-n increases.

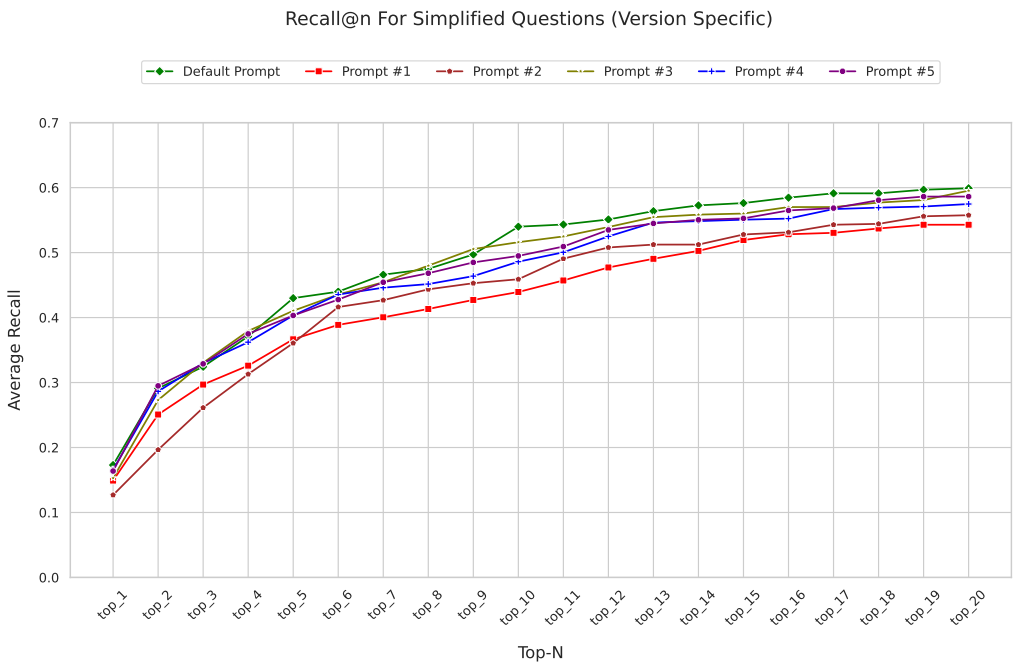

Figure 5.7: *Recall@n for simplified questions (version-specific) using different prompts*

For simplified questions version-specific case (Fig. 5.7), similar to non-version-specific case, Prompt #2 starts with the lowest recall. Prompt #1 and #2 consistently remain lowest across all the values of n. The recall for the default prompt, Prompt #3, #4 and #5 closely follow each other, with minor fluctuations, up to the top-9 documents, after which the default prompt consistently stays above other prompts. Figure 5.7 reveals that the performance increases rapidly and consistently up to the top-15 documents after which the recall starts to saturate around top-17 across all the prompts.

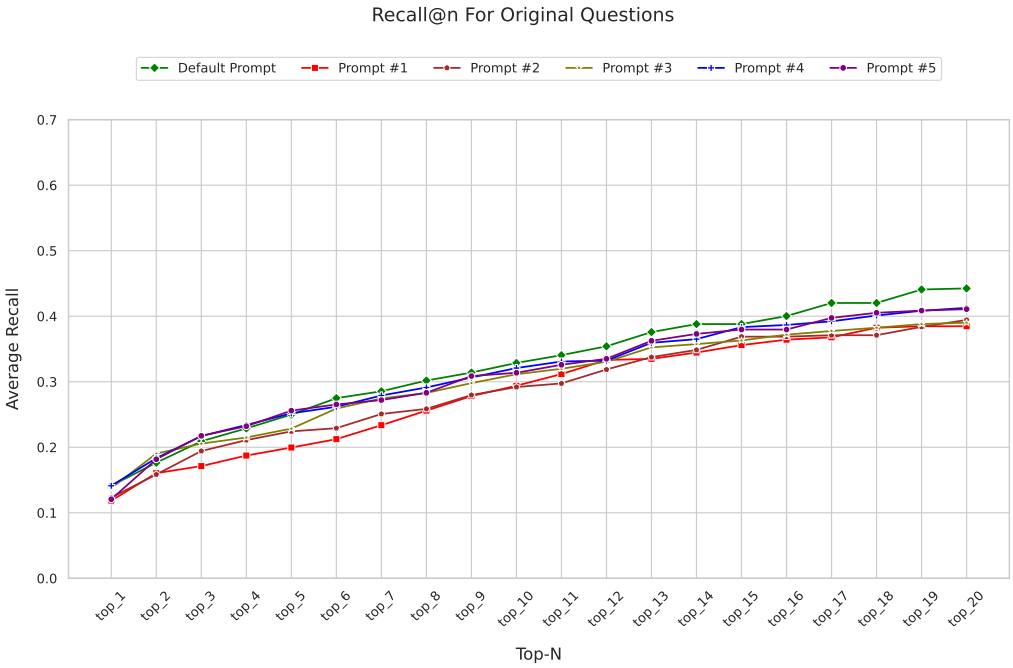

Figure 5.8: *Recall@n for original questions using different prompts*

Original questions in the non-version-specific case (Fig. 5.8) show a similar near-linear trend as simplified questions in the non-version-specific results (Fig. 5.6). The difference can be seen on the y-axis, where the overall recall for original question is lower than simplified questions across all the prompts except Prompt #2. Interestingly, the recall with Prompt #2 for original questions stays above the recall for simplified questions across all values of n. The lines in Figure 5.8 are closely grouped for the original questions, whereas they are more dispersed for the simplified questions. Overall, the default prompt still seems to beat the other prompts especially for higher values of n.

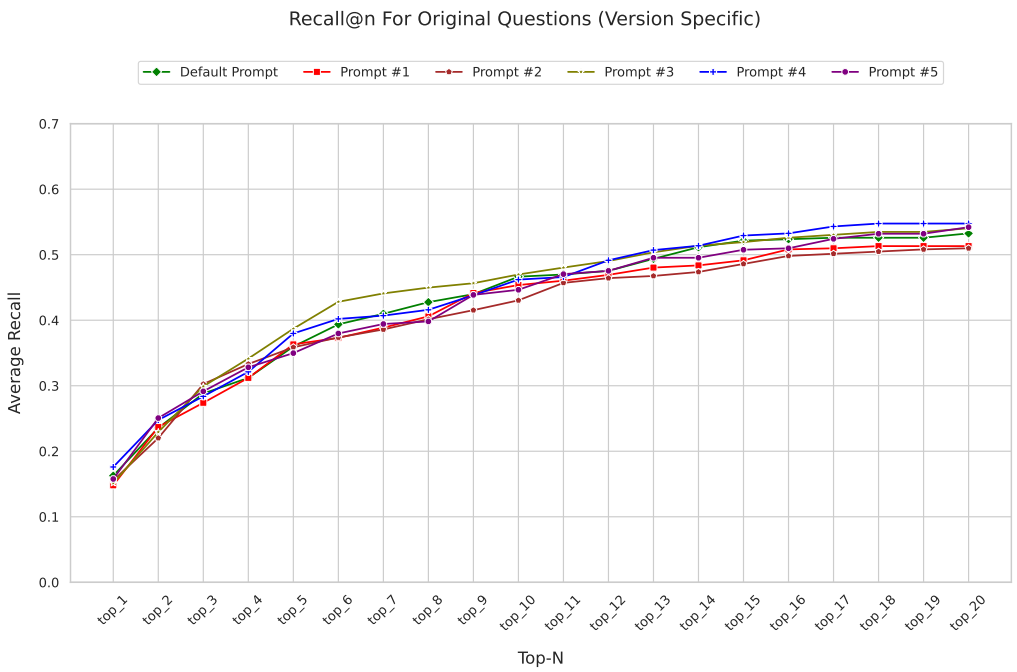

Figure 5.9: *Recall@n for original questions (version-specific) using different prompts*

Original questions in the version-specific case (Fig. 5.9) follow the same trend as simplified questions in the version-specific case (Fig. 5.7). For smaller values of n, i.e. top-4 to top-11, Prompt #3 achieves the highest performance. After top-12, Prompt #4 yields the best performance throughout the end of the plot. For this particular case, more descriptive prompts (#3, #4 and #5) leads to better performance than the default prompt. It can also be noticed that the lines are clustered even more closely together in this case. A clear saturation in performance around top-17 documents, similar to (Fig. 5.7), can also be seen in this plot.

Concluding all 4 recall plots, the results show a clear distinction in the behavior of recall scores between original and simplified questions. For simplified questions, the scores are more scattered, indicating a higher variance in recall across different prompts. On the other hand, for original questions, the scores are less deviated,

suggesting that different prompts have less impact on recall. Upon comparing the generated keywords, it seems that prompts leading to longer lists of keywords tend to perform poorly compared to those producing short but concise keywords. For instance, Prompt #2 specifically instructs the model to include "synonyms or related terms to capture a broader range of relevant information," whereas other prompts direct the model to generate keywords that are concise and relevant.

Average number of keywords generated by each prompt		
Prompt	**Simplified Question**	**Original Question**
Default	7.10 keywords	9.86 keywords
#1	12.86 keywords	14.36 keywords
#2	15.21 keywords	15.79 keywords
#3	7.80 keywords	9.33 keywords
#4	7.52 keywords	9.65 keywords
#5	7.54 keywords	9.75 keywords

Prompt #1 and #2 produce more keywords on average as compared to the other prompts. These two prompts also lead to lowest recalls and MRR in all the cases. In some cases, incorporating synonyms and additional related terms improves the rankings, yet the overall results are suboptimal. This indicates that longer keywords might make the query more specific, but they mostly introduce noise if the keywords are not well-matched with the documents.

6 Conclusion

This aim of this research was to evaluate the performance and effectiveness of a RAG-based system, with a specific focus on the retriever component. A crucial part of this thesis involved the creation of a comprehensive dataset suitable for various experiments conducted throughout the study. The dataset includes a wide variety of customer queries across various software products. The second part consists of experimentation and analysis to address the key research questions.

With respect to the first research question, the impact of keyword generation/extraction process on the system was assessed. The results show that keywords generation has a positive impact on performance. The inclusion of this step led to more accurate document retrieval for both simplified and original questions. While the impact was much higher for simplified questions, the results of original questions also demonstrated a positive result. This can be attributed to the complexity and more descriptive nature of the original questions. Overall, it can be concluded that omitting the keyword extraction process would likely result in a decline in the system's performance.

With respect to the second research question, the study explored the effects of different prompting techniques on the keyword extraction process. Some prompts were designed to include additional synonyms and related terms because the BM25 algo-

rithm, which is used by the search engine, does not consider the semantic meaning of the words. Therefore, prompts with additional terms were designed to capture a wide range of potential matches. However, this approach often introduced noise and reduced overall retrieval performance. It can be concluded that prompts that generated fewer, more targeted keywords outperformed the prompts designed to include additional terms.

Finally, the results also reveal that including the version of the software product in the search query significantly improves the retrieval results and leads to higher rankings of the relevant documents. Figure 5.2 and 5.3 reveal that specifying the version without generating keywords (the dotted green line) outperforms the non-version-specific case with keyword generation (the solid blue line). This suggests that while keyword generation does improve performance, including the version of the software in the query has a significantly higher positive impact on performance. The recall plots also indicate a saturation of performance around the top-17 documents for the version-specific case, whereas the performance for the non-version-specific case doesn't seem to saturate. Optimally, the system should achieve the best performance when the software version is specified, keywords are generated and approximately top 15 documents are used to generate a response.

7 Future Work

The findings in this study establish the foundation for future work aimed to further evaluate and enhance the performance of the existing system. The experiments reveal that simplified version of the questions yields better results than original user questions. This opens up possibilities for further research by incorporating query rewriting models in the system. Research suggests that query rewriting can improve the way user's search queries are phrased which in turn improves the search results. Ma et al. [32] introduce a Rewrite-Retrieve-Read framework that focus on improving the search query before performing the retrieval and augmentation steps. In order to address the limitation of the current retrieval algorithm, which fails to consider the semantic meaning of the query, the future research could explore the integration of more advanced dense-vector retrieval techniques such as neural network-based retrieval models. Integrating the models that consider the semantic meaning of the query and the documents may lead to more precise retrieval of relevant information. The solutions provided by the support team during direct interaction with customers can be documented and used to supplement the knowledge base. This data can be used to address the issues that may not be answered by the existing documentation data. This also opens up an opportunity to expand the evaluation dataset. Furthermore, a recent study on Conversational QA Models [33] introduces innovative approaches that can be utilized to address complex customer support scenarios, which requires multiple back-and-forth interactions to resolve.

References

[1] J. Haase and P. H. Hanel, "Artificial muses: Generative artificial intelligence chatbots have risen to human-level creativity", *Journal of Creativity*, vol. 33, no. 3, p. 100 066, 2023.

[2] E. Stoilova, "Ai chatbots as a customer service and support tool", *ROBONOMICS: The Journal of the Automated Economy*, vol. 2, pp. 21–21, 2021.

[3] M. A. Camilleri and C. Troise, "Live support by chatbots with artificial intelligence: A future research agenda", *Service Business*, vol. 17, no. 1, pp. 61–80, 2023.

[4] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training", in *International conference on machine learning*, PMLR, 2020, pp. 3929–3938.

[5] A. Roberts, C. Raffel, and N. Shazeer, "How much knowledge can you pack into the parameters of a language model?", *arXiv preprint arXiv:2002.08910*, 2020.

[6] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks", *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[7] T. Shen, X. Geng, C. Tao, *et al.*, "Unifier: A unified retriever for large-scale retrieval", in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4787–4799.

[8] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries", in *Proceedings of the first instructional conference on machine learning*, Citeseer, vol. 242, 2003, pp. 29–48.

[9] S. Qaiser and R. Ali, "Text mining: Use of tf-idf to examine the relevance of words to documents", *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, 2018.

[10] I. A. Heggo and N. Abdelbaki, "Behaviorally-based textual similarity engine for matching job-seekers with jobs", in *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, Springer, 2018, pp. 564–574.

[11] S. Robertson, H. Zaragoza, *et al.*, "The probabilistic relevance framework: Bm25 and beyond", *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[12] V. K. Singh and V. K. Singh, "Vector space model: An information retrieval system", *Int. J. Adv. Engg. Res. Studies/IV/II/Jan.-March*, vol. 141, no. 143, 2015.

[13] R. Subhashini and V. J. S. Kumar, "Evaluating the performance of similarity measures used in document clustering and information retrieval", in *2010 first international conference on integrated intelligent computing*, IEEE, 2010, pp. 27–31.

[14] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, "A survey on retrieval-augmented text generation", *arXiv preprint arXiv:2202.01110*, 2022.

[15] F. Petroni, P. Lewis, A. Piktus, *et al.*, "How context affects language models' factual predictions", *arXiv preprint arXiv:2005.04611*, 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need", *Advances in neural information processing systems*, vol. 30, 2017.

[17] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training", 2018.

[18] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, "Challenges and applications of large language models", *arXiv preprint arXiv:2307.10169*, 2023.

[19] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in conversation", *arXiv preprint arXiv:2104.07567*, 2021.

[20] M. A. C. Soares and F. S. Parreiras, "A literature review on question answering techniques, paradigms and systems", *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 6, pp. 635–646, 2020.

[21] T. Kwiatkowski, J. Palomaki, O. Redfield, *et al.*, "Natural questions: A benchmark for question answering research", *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019.

[22] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension", *arXiv preprint arXiv:1705.03551*, 2017.

[23] S. Rosenthal, A. Sil, R. Florian, and S. Roukos, "Clapnq: Cohesive long-form answers from passages in natural questions for rag systems", *arXiv preprint arXiv:2404.02103*, 2024.

[24] P. Bajaj, D. Campos, N. Craswell, *et al.*, "Ms marco: A human generated machine reading comprehension dataset", *arXiv preprint arXiv:1611.09268*, 2016.

[25] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models", *arXiv preprint arXiv:2104.08663*, 2021.

[26] J. White, Q. Fu, S. Hays, *et al.*, "A prompt pattern catalog to enhance prompt engineering with chatgpt", *arXiv preprint arXiv:2302.11382*, 2023.

[27] S. Ekin, "Prompt engineering for chatgpt: A quick guide to techniques, tips, and best practices", *Authorea Preprints*, 2023.

[28] L. Giray, "Prompt engineering with chatgpt: A guide for academic writers", *Annals of biomedical engineering*, vol. 51, no. 12, pp. 2629–2633, 2023.

[29] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners", *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[30] Amazon, *Amazon opensearch service*, Accessed: January 28, 2024, 2024. [Online]. Available: `https://aws.amazon.com/opensearch-service/`.

[31] OpenAI, *Openai models*, Accessed: February 5, 2024, 2024. [Online]. Available: `https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo`.

[32] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting for retrieval-augmented large language models", *arXiv preprint arXiv:2305.14283*, 2023.

[33] Z. Liu, W. Ping, R. Roy, P. Xu, M. Shoeybi, and B. Catanzaro, "Chatqa: Building gpt-4 level conversational qa models", *arXiv preprint arXiv:2401.10225*, 2024.