International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

A Document Retrieval System Using Retrieval-Augmented Generation

Rayudu Bharath Kumar
Reg. No. 24Q71F0051
rbharathkumar658@gmail.com
Department of Master of Computer Applications
Avanthi Institute of Engineering and Technology (Autonomous)
Vizianagaram, Andhra Pradesh, India
Under the guidance of Mr. Ch. Bhupathiraju, MCA, Assistant Professor
challa.bhupati@gmai.com

***Abstract*—*In the age of exponential information growth, effective document retrieval has become essential for knowledge-based systems. This project presents a Document Retrieval System using Retrieval-Augmented Generation (RAG), an architecture that combines traditional retrieval mechanisms with the generative power of transformer-based language models. RAG enhances the retrieval process by dynamically integrating external documents into the response-generation pipeline, allowing the system to produce more accurate and contextually relevant answers. The system first retrieves relevant passages from a pre-indexed document corpus using vector-similarity search (FAISS), and the retrieved documents are then passed to a generative language model that synthesises the final output based on both the input query and the retrieved content. This hybrid approach ensures factual grounding while maintaining the flexibility of generative models, and it reduces hallucinations by conditioning generation on retrieved context. The solution is implemented in Python using LangChain, FAISS, and transformer-based language models, and is particularly effective for question answering, summarisation, and knowledge-base augmentation. Testing and evaluation across document processing, semantic retrieval, response generation, scalability, and reliability confirmed that the system retrieves relevant information accurately and generates context-aware responses. The approach demonstrates improved performance over traditional retrieval-only or generation-only models, offering a scalable and intelligent document-access system suitable for enterprise, academic, and personal use.*

Keywords*—*Retrieval-Augmented Generation; Document Retrieval; Vector Embeddings; Semantic Search; FAISS; Large Language Models; Natural Language Processing; LangChain.

I. INTRODUCTION

In today's data-driven world, organisations are inundated with vast volumes of unstructured textual information—documents, emails, reports, and knowledge bases. Efficiently retrieving relevant information from this data is critical for decision-making, customer service, and research. Traditional keyword-based search systems often fall short in understanding context, semantics, and user intent.

To overcome these limitations, Retrieval-Augmented Generation (RAG) has emerged as a powerful architecture that combines the strengths of retrieval-based and generation-based models. A document

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

retrieval system using RAG enhances the accuracy and relevance of responses by first retrieving the most pertinent documents from a corpus and then generating a natural-language answer based on that retrieved content. The architecture leverages pretrained language models for retrieval and transformer-based models for generation, enabling open-domain question answering and document querying with high contextual understanding.

This project aims to design and implement a Document Retrieval System using RAG, showing how combining dense retrieval techniques with generative models can deliver intelligent, context-aware, and concise responses to user queries. The hybrid approach addresses the shortcomings of isolated retrieval or generation systems and is particularly useful in domains where information is scattered across multiple sources.

II. LITERATURE SURVEY

Document retrieval systems have evolved significantly with advances in information retrieval (IR), natural language processing (NLP), and artificial intelligence. Traditional IR systems focused on keyword-based searching such as Boolean search, TF-IDF, and BM25; these were effective for basic searching but often failed to understand the semantic meaning and context of user queries, producing irrelevant or incomplete results. To improve retrieval, researchers introduced semantic-search techniques using word embeddings and transformer-based models such as BERT, which understand contextual relationships between words and sentences and significantly improve retrieval quality.

With the growth of Large Language Models (LLMs), conversational AI systems became capable of generating human-like responses, but standalone language models depended mainly on pretrained knowledge and sometimes produced inaccurate or hallucinated information. Retrieval-Augmented Generation was introduced to address this by combining information retrieval with LLMs: relevant document content is retrieved from a knowledge base and provided as context to the language model, reducing hallucinations and increasing reliability. Studies have demonstrated the effectiveness of vector databases and embedding-based similarity search; tools such as FAISS are widely used for storing and retrieving embeddings efficiently, and frameworks such as LangChain simplify the integration of language models with retrieval pipelines. RAG-based systems show strong performance in question answering, enterprise document search, customer support, and knowledge management.

TABLE I. EVOLUTION OF DOCUMENT RETRIEVAL APPROACHES

S.No	Approach	Technique	Limitation / Note
1	Classical IR	Boolean, TF-IDF, BM25	No semantic/context understanding
2	Semantic search	Word embeddings, BERT	Improved contextual matching
3	Large Language Models	Transformer generation	Hallucination; pretrained-only knowledge

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

S.No	Approach	Technique	Limitation / Note
4	Dense passage retrieval	DPR-style retrievers	Strong open-domain retrieval
5	Retrieval-Augmented Generation	Retriever + generator	Grounded, context-aware responses
6	Vector databases	FAISS similarity search	Fast, scalable embedding retrieval

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

Existing document-retrieval systems mainly rely on traditional keyword-based search to retrieve information from large document collections, using methods such as Boolean search, pattern matching, and basic database queries. These methods are simple but often fail to understand the meaning and context of user queries, producing irrelevant results. Retrieval depends heavily on manually defined keywords and indexing, so if a query does not contain the exact keywords present in the documents, relevant information may not be retrieved even when related content exists. Many systems return only document lists or links without generating meaningful responses, and they struggle with scalability and retrieval performance as collections grow.

Disadvantages of the existing system:

- Keyword-based searching often produces irrelevant or inaccurate results.
- Cannot effectively understand semantic meaning or query context.
- Exact keyword matching fails on synonyms and paraphrases.
- Returns document lists/links without meaningful responses or summaries.
- Limited scalability and slower retrieval on very large collections.

B. Proposed System

The proposed system is a Document Retrieval System using RAG that combines semantic document retrieval with advanced language generation to provide context-aware, meaningful responses. It consists of modules for document uploading, text preprocessing (cleaning, tokenisation, normalisation), embedding generation, vector-database storage using FAISS, semantic retrieval, and response generation. Uploaded documents are split into chunks and converted into vector embeddings that capture semantic meaning; when a user submits a query, it is embedded and compared with stored vectors to retrieve the most relevant chunks by semantic similarity rather than exact keyword matching. The retrieved chunks are passed as context to a language model that generates an accurate, human-readable answer.

Advantages of the proposed system:

- Improved retrieval accuracy through semantic similarity search.
- Faster response time using optimised FAISS indexing.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

- Better contextual understanding of documents and queries.
- Scalable to large document collections.
- Intelligent, context-aware response generation that reduces hallucinations.
- Applicable across education, research, healthcare, and enterprise knowledge management.

IV. SYSTEM DESIGN AND METHODOLOGY

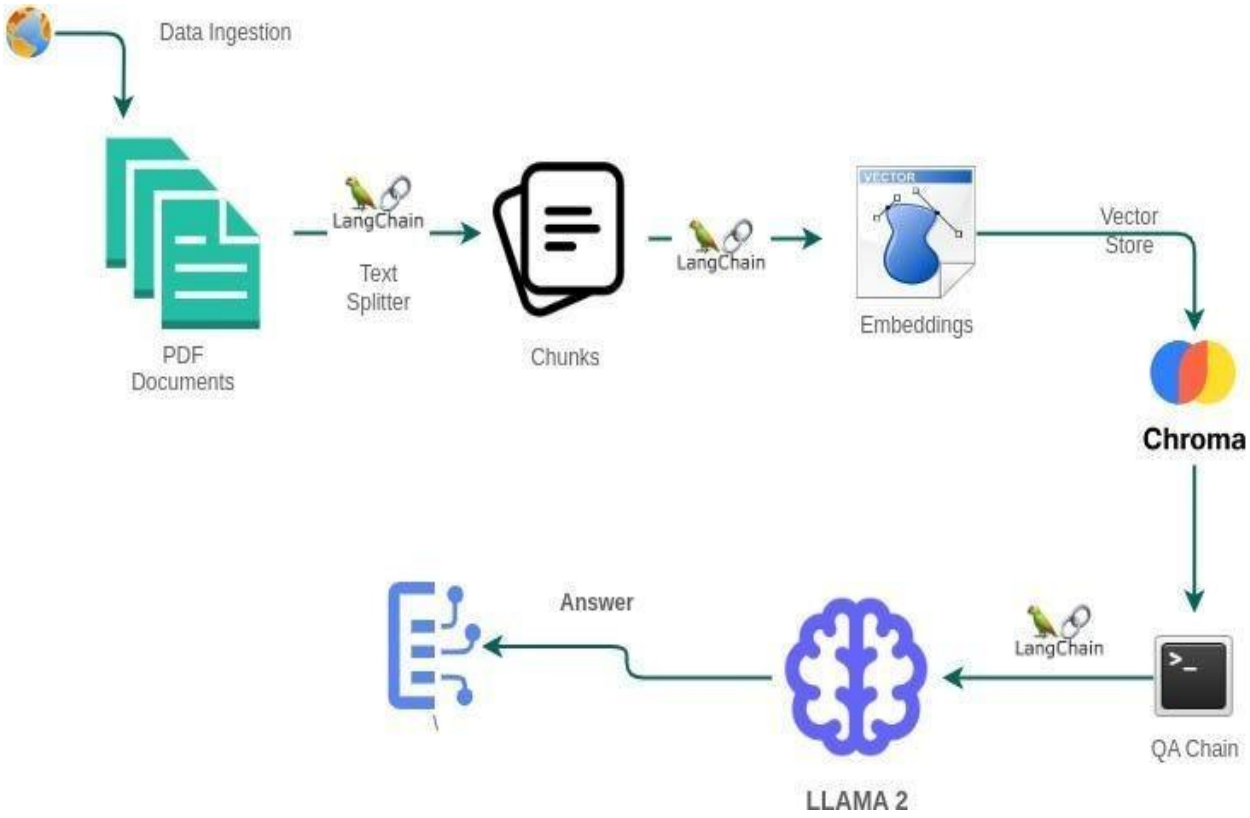
A. Methodology

The system combines document retrieval with Large Language Models to improve the quality and relevance of generated answers. Documents such as PDFs, text files, and reports are collected as the knowledge base and preprocessed to remove unnecessary symbols, extra spaces, and formatting issues. The documents are divided into smaller text chunks for efficient retrieval, and each chunk is converted into a vector embedding that represents its semantic meaning. The embeddings are stored in FAISS, which acts as a vector database for fast similarity search. When a user enters a query, it is preprocessed and embedded with the same model, compared against the stored embeddings to identify the most relevant chunks, and those chunks are provided to the language model as additional context to generate an accurate, context-aware response that is displayed in the application interface.

B. System Architecture

The architecture uses a dual approach: a retriever that locates the top-k relevant documents using vector similarity (FAISS), and a generator that conditions on the retrieved documents to generate a coherent response using a language model. Around these core components, the system provides document-upload, preprocessing, embedding-generation, vector-storage, retrieval, and response-generation modules, integrated through a pipeline so that retrieval and generation operate together rather than in isolation, addressing the disconnect between retrieval and generation in pretrained-only models.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper



C. Functional Behaviour

Functionally, the system must accept and process uploaded documents, preprocess and chunk text, generate and store embeddings in FAISS, accept user queries, perform semantic retrieval of relevant chunks, generate context-aware responses, and display them through the interface. It must also validate inputs and handle unsupported document formats gracefully so that invalid inputs do not cause crashes.

V. SYSTEM IMPLEMENTATION

A. Technology Stack

TABLE II. TECHNOLOGY STACK

Component	Technology / Tool
Programming Language	Python
Orchestration Framework	LangChain
Vector Database	FAISS
Language Models	Transformer-based LLMs (e.g., BERT/BART/T5-style models)
Embeddings	Embedding models for semantic vector representation

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Component	Technology / Tool
Document Types	PDF, text files, reports
Application Interface	Web-based user interface

B. Implementation Pipeline

The implementation realises the methodology as an integrated pipeline. The document-processing module extracts and cleans text and divides it into chunks. The embedding module converts chunks into vector representations, which the storage module indexes in FAISS for fast similarity search. The retrieval module embeds the user query and fetches the most semantically similar chunks. The generation module passes the retrieved context together with the query to the language model to produce a context-aware answer, which the interface module returns to the user. The system is developed in Python with LangChain coordinating the retrieval and generation components.

C. Retrieval and Generation

Retrieval is performed by comparing the query embedding with stored document embeddings and selecting the top-k most similar chunks, so that contextually related information is fetched even when the query wording differs from the document text. Generation conditions the language model on the retrieved chunks, which grounds the output in actual source content and reduces hallucination compared with generation-only models. This combination yields responses that are both fluent and factually anchored to the corpus.

VI. SYSTEM TESTING AND RESULTS

The system was tested extensively after implementation and integration of all modules—document uploading, preprocessing, embedding generation, FAISS indexing, semantic retrieval, and response generation. Different documents and datasets were used to examine functionality under various conditions. Representative functional test cases covered the login form (validating the user against the database), the user-registration form (checking that all fields are completed before saving), and the change-password function (validating the old and new password fields and showing an error on mismatch). The reported results indicate the tested functions behaved as expected.

TABLE III. REPRESENTATIVE TEST CASES

ID	Function	Expected Result	Status
TC01	Login	Validate the user and check existence in the database	**Pass**
TC02	User registration	Check all fields are filled and save the user	**Pass**
TC03	Change password	Validate old/new password fields; error on mismatch	**Pass**

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

A. Results and Evaluation

The system was evaluated on retrieval accuracy, response time, scalability, and reliability. Uploaded documents were processed correctly through extraction and preprocessing, and the embedding model successfully converted text into vector representations stored efficiently in FAISS, which enabled faster retrieval even as the dataset grew. For multiple user queries, the retrieval module identified and fetched the most relevant documents based on semantic similarity rather than keyword matching, improving accuracy and context understanding, and the generation module produced relevant, context-aware answers. Scalability testing showed stable retrieval efficiency on large collections, and reliability testing confirmed consistent operation with effective handling of invalid inputs and unsupported formats. The source describes these outcomes qualitatively; no specific numeric scores are claimed here.

Representative screenshots from the prototype implementation:

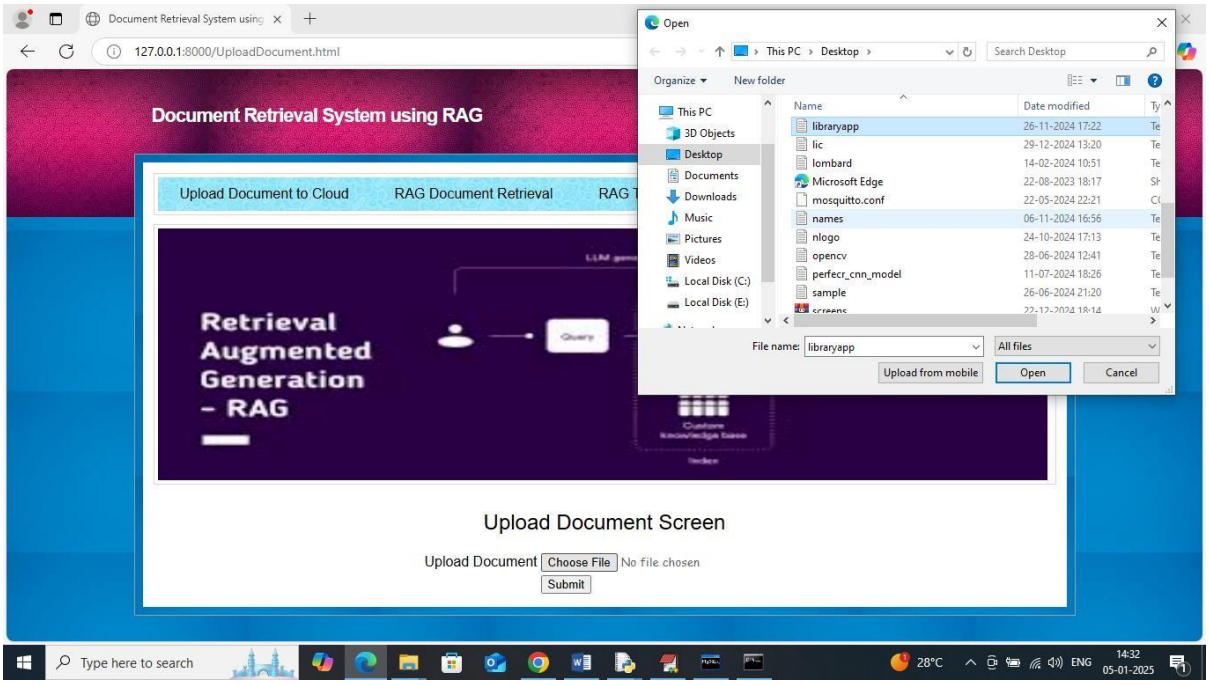


Fig. 1. Document upload and preprocessing.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

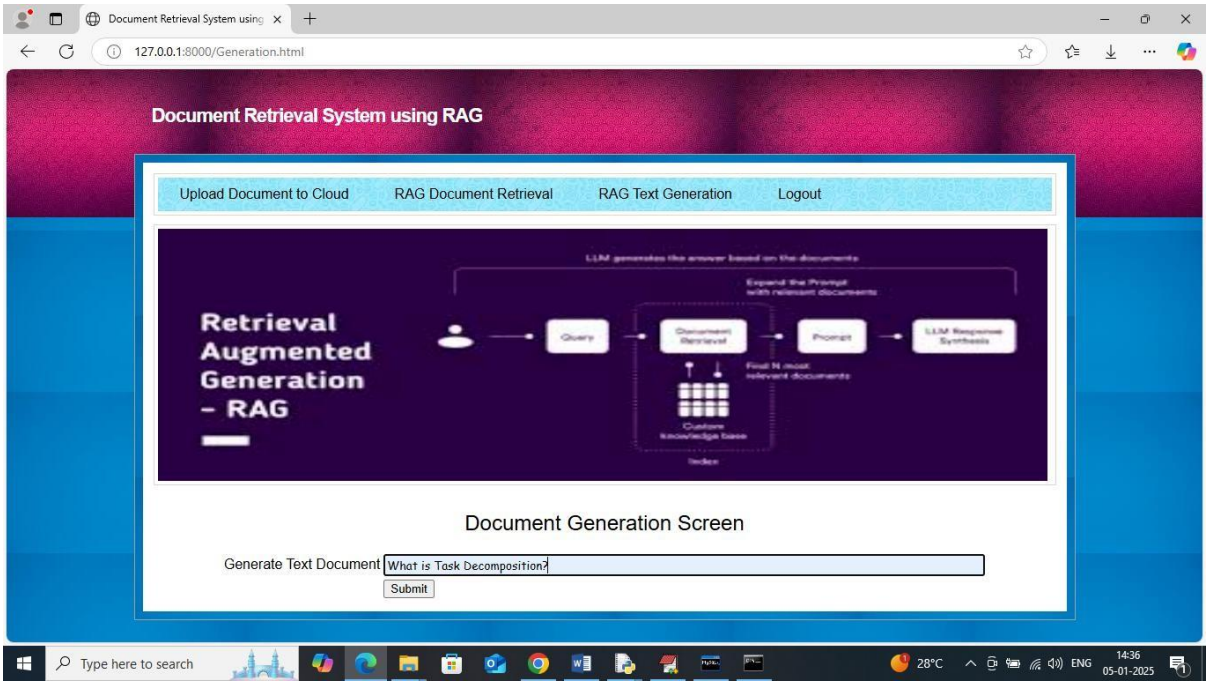


Fig. 2. Semantic retrieval for a user query.

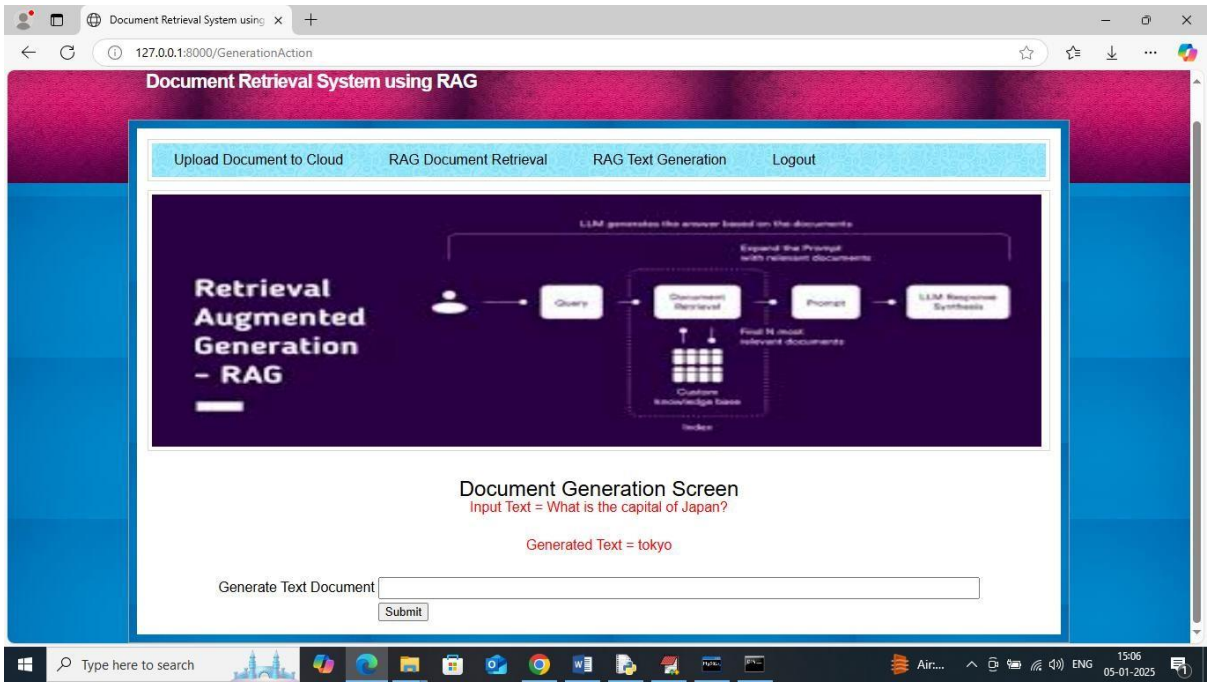


Fig. 4. Context-aware generated response.

VII. CONCLUSION AND FUTURE SCOPE

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

The Document Retrieval System using RAG was successfully designed and implemented to provide an efficient and intelligent solution for retrieving information from large document collections. By integrating semantic retrieval with advanced language generation, the system delivers accurate, context-aware, and meaningful responses, overcoming the limitations of keyword-based retrieval. FAISS indexing improved retrieval speed and allowed the system to handle large datasets with reduced search time, and the RAG approach enhanced response quality by generating answers from retrieved contextual information. Testing and evaluation confirmed that the system processes different document formats, generates relevant responses, scales to large datasets, and operates reliably with effective error handling.

There are several opportunities for future enhancement. The system can be extended to support multilingual document retrieval, integrate larger and more advanced language models, improve retrieval-ranking algorithms, and implement real-time document updates. Cloud deployment and distributed storage can improve accessibility, scalability, and performance for enterprise-level applications, while additional security features and user-authentication mechanisms can further enhance reliability and data protection.

REFERENCES

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

[2] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in Proc. EMNLP, 2020.

[3] A. Y. Ng and M. I. Jordan, "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes," in Advances in Neural Information Processing Systems (NIPS), 2002.

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL, 2019.

[5] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research, 2020.

[6] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," in Proc. ACL, 2017.

[7] K. Lee, M. Chang, and K. Toutanova, "Latent Retrieval for Weakly Supervised Open Domain Question Answering," in Proc. ACL, 2019.

[8] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in Proc. ICML, 2020.

[9] M. Chen et al., "Multimodal Retrieval-Augmented Generation for Knowledge-Intensive Tasks," arXiv preprint, 2021.

[10] S. Yao et al., "Efficient QA with Dense Retriever and Lightweight Generator," in Findings of EMNLP, 2021.