RESEARCH PAPER

Citechat: A Rag Based Intelligent Chatbot

Suganya R[1], Kavitha V[2], Shereya B[3], Sujitha C[4], Dharani G[5], Banu Rithika M S[6]

[1]Assistant Professor, Department of Artificial Intelligence and Data Science, V.S.B. Engineering College, Karur, India.
Email: suganyavsb20163@gmail.com

[2]Assistant Professor, Department of Artificial Intelligence and Data Science, V.S.B. Engineering College, Karur, India.
Email: kavithataru2015@gmail.com

[3]Department of Artificial Intelligence and Data Science, V.S.B. Engineering College, Karur, India.
Email: shereyabaskaran138@gmail.com

[4]Department of Artificial Intelligence and Data Science, V.S.B. Engineering College, Karur, India.
Email: sujithachandrasekaran2004@gmail.com

[5]Department of Artificial Intelligence and Data Science, V.S.B. Engineering College, Karur, India.
Email: dharanigokulakannan962004@gmail.com

[6]Department of Artificial Intelligence and Data Science, V.S.B. Engineering College, Karur, India.
Email: banurithikabanurithika@gmail.com

***Abstract*—**CiteChat is a document-based intelligent chatbot, which uses Retrieval-Augmented Generation (RAG) to provide correct and context-sensitive answers of documents stored by users, including resumes, research papers and articles. It will take into account the fact that the proposed system will offer more factual consistency as compared to conventional large language model (LLM) systems because it will base every one of its responses on the uploaded document. A stringent grounding procedure is put in place to avoid the production of unsupported answer generation by introducing similarity threshold validation which minimizes hallucination. First, the PDF documents are read with the help of transformer-based sentence embeddings, and the embeddings are stored in the local memory, preserving the data privacy but allowing the data to be easily retrieved. When the user enters a query, the semantic similarity search mechanism finds the most relevant document segments and the most relevant document segments are made available to a Groq-powered LLaMA model to generate the response. The system also increases the level of transparency by including page-level snippets of citations and generated answers along with calculation of a retrieval-based confidence score to show the reliability of answers. Experimental analysis shows better factual basis and fewer hallucination in contrast to the standalone approaches based on the LLM. Lastly, the lightweight Streamlit interface helps to upload documents and interactively query, making it easy to analyze documents.

Keywords*—Document retrieval, large language model, Retrieval Augmented Generation, Vector Database, Hallucination.*
How to cite this article: Suganya R, Kavitha V, Shereya B, Sujitha C, Dharani G, Banu Rithika MS. CiteChat: A RAG Based Intelligent ChatBot. Int J Drug Deliv Technol. 2026;16(13s): 975-985. DOI: 10.25258/ijddt.16.13s.109

I. Introduction

Artificial Intelligence, or AI, is a fundamental technology of digital change in recent years and the area of education, healthcare, banking, hiring, and business data management. AI refers to the computer systems which attempt to behave in a manner similar to the way human beings reasoning. They are able to learn with data, ponder on issues and reach conclusions. Generative Artificial Intelligence is one of the branches of AI that has increased significantly in the recent years. Generative AI is used to generate new information such as text, images, and even code, through its learning patterns on large volumes of data. The key representative of this method is Large Language

Models (LLMs) that drive most of the modern chat [11], [13]. They are trained on transformer based architectures like BERT [11] and subsequent generative models including GPT and LLaMA.

Chatbots are the most widespread applications of Generative AI nowadays. They enable individuals to interact with machines by use of normal language. Without the assistance of a human operator, these systems are able to provide fast answers. A great number of them are applied in the support of studies, customer service, reading documents, and office work. Nevertheless, older chatbot systems rely largely on what the model learnt in the course of training. They never bother to check whether the answer is present in the user document or not. Due to this fact they can provide false information or even fabricate information. This issue is critical when proper facts and document evidence are highly needed.

To deal with this challenge, Retrieval-Augmented Generation (RAG) was proposed as a powerful model of improving document-based question answering [4]. RAG comprises two primary elements, namely a retrieval hand that extracts pertinent information in the external documents and a generation hand that generates the answers using the extracted information. Rather than searching using pretrained knowledge, the system initially tries to find the relevant passages of the document, and this is followed by generation of responses based on the information. RAG can diminish chances of a text being hallucinated or unsupported by conditioning the generation on retrieved text and enhances

transparency [4], [7]. The dense passage retrieval assists the system to match the system more accurately with the question and document meaning [5]. The use of models such as Sentence-BERT transforms text into vectors in order to easily compute similarity [6]. The retrieval-based mechanisms are broadly used in the modern document chat systems to increase the quality of answers and ground them in context [7], [8].

It is built on these concepts that this research introduces CiteChat, an assistant chat that is document-oriented. It is able to read PDF files uploaded by users, and provide context-based responses. The system reads the text of the file and divides it into small chunks which are then converted to embeddings. Such embeddings are embedded in a local vector database to search. Similarity search is used to find the most related text parts when a user poses a question to the system. These sections are then given to a big language model to provide an answer that is clear and supported. Citations are also displayed on the system and a confidence score is provided so that the user is able to know the reliability of the answer.

Although Generative AI and RAG methods are superior to the previous designs of chatbots, there are still certain issues. The ability to fully control the incorrect answers, secure the user data, and work with numerous documents simultaneously is still challenging. The latest experiments on retrieval mechanisms identify the issues of hallucination management and accuracy of retrieval [7], [9]. Such open issues indicate that a structured document-style chatbot, with semantic search, citation assistance, and secure local storage of embeddings is needed.

That is why the following section describes the problem statement in detail. It explains the research gaps existing in the existing systems and the reason behind why a credible document-based chat system is required.

II. Problem Statement

Reliable document-based question answering is a difficult research issue despite good advances in conversational AI and document tools. Retrieval-augmented models have also been utilized in many studies lately to enhance the accuracy of answers in document-based systems [7]. Multi-document retriever and reader models were also introduced in some of the works, to select improved passages and base the final answer on them [3]. Nevertheless, semantic understanding, factual correctness, transparency, confidence scoring, and data privacy are hard to manage simultaneously in most of the modern systems.

The traditional method of search with keys works only in the situation when the exact words or very similar ones are observed both in the query and in the document. The query used by the user is q and the document is d. A keyword system only finds results when there is overlap of words:

$$q \cap d \neq \phi \quad (1)$$

In this case ∩ represents shared words between the query and the document. This approach does not work when the same meaning is expressed in various words. Thus, valuable information will not be accessed, particularly long or intricate files. The introduction of dense retrieval solutions to this meaning gap problem occurred [5]. However, the still most common document systems rely primarily on keyword matching.Big language models obtain responses through probability-based text prediction. They do not necessarily examine the document itself. The answer to a pure generative model may be expressed as:

$$R=LLM(q) \quad (2)$$

In this case, R represents the response generated and q means the question asked. The model relies on its training data, which means the result might not be a genuine one based on the uploaded document. This has the potential to minimize factual accuracy that has been extensively talked about in retrieval studies [4], [9]. Such unsubstantiated answers lead to lower levels of trust in an academic and professional context.This is enhanced by retrieval-augmented generation which relies on the retrieved document content in order to come up with the response [4]. Even then, some issues remain. In embedding-based systems, query embedding vector eq and document embedding vector "e" ("d" "i") are often compared with the help of a cosine similarity measure [5], [6].In this case, d is a single segment of a document. This computation assists in causing a match of sense rather than precise words. Nevertheless, several systems fail to display similarity scores or level of confidence to users. Due to this, the process is also ambiguous and difficult to check.Another issue is privacy. Cloud services are used by many document chatbots in the creation and storage of vectors. In case of uploading of private files like resumes, marksheets, or company report, transmission to third party servers may create concerns about data. Big vector search libraries scale and are fast to retrieve [12]. Nevertheless, local storage of embeddings in the name of privacy is not widespread in most systems in the real world.Issue of lack of citation support also exists. Users usually are not able to observe whether or not the answer is supported by the text of the document. Despite the fact that there is some research on the topic of hallucination identification within the framework of retrieval-based systems [9], most of the tools do not provide the snippets of the sources or information on the pages. No evidence can be easily checked by the users.With these gaps, the following research question will be the primary one in this study:

How might a conversational AI system be able to produce semantically valid and document-based answers and at the same time offer citation visibility, confidence ranking, and privacy-compliant document processing?

The solution to this would be to use a system that needs to integrate embedding-based retrieval [6], dense similarity calculation [5], retrieval-augmented generation [4], citation-

supported validation and local embedding storage into a single transparent architecture.

The following part examines the available document analysis systems in more detail to learn more about these practical constraints.

III. Existing System

A. Overview of Existing System

Majority of the existing document analysis procedures rely on manual reading and search by keywords. The users typically access PDF files and attempt to locate information using default search applications. This applies to resumes, research papers and reports. These tools are replicant of exact words. They are really not aware of what the question is about. The failure of the keyword search in the densely retrieved studies as expounded in dense retrieval [5] involves writing the same idea in varying words.The other popular strategy is the use of chatbot systems that are driven by large language models that do not add any new data to their function but produce responses solely on the basis of the training data. Such systems can even fail to check the existence of the answer generated in the documents provided. The first proposal of retrieval-augmented frameworks was made to enhance the knowledge grounding and contextual usage of language models [4], [10]. Nonetheless, numerous practical applications do not have a high level of citation support and clear transparency processes [7].

Most document chatbot models rely on searching libraries based on vectors to index and to search by similarity. Scalable, efficient, and efficient storage and retrieval of vectors is possible using libraries like FAISS [12]. Even though these systems enhance performance and processing speed, in real-life applications embeddings can be stored on remote or external servers where their privacy may be questioned when working with sensitive documents. The majority of the existing systems also lack the clear citation evidence and confidence ratings. Hallucinations-detection research has been done in retrieval-based models [9], but are not commonly added to practical document tools.

B. Workflow of Existing System

In the traditional document systems, the routine of the work is very easy:

1) The user opens or uploads an item or a document.
2) Searching of keyword is done by the user.
3) The system displays similar locations of words.
4) The results are manually read and interpreted by the user.
5) A chatbot will provide answers in certain situations and will not look at document grounding.

This is slow and inefficient when dealing with large files, particularly to deal with complicated queries. This can be automated by multi-document retriever and reader model that endeavors to retrieve and read multiple documents together [3]. The process of retrieval and generation of the answer is however loosely intertwined in most systems.

C. Performance Comparison of Existing Methods

TABLE I. Existing System Performance Comparison

Method	Semantic Search	Citation Support	Reliability	Privacy
Manual search	No	No	Medium	High
PDF keyword search	No	No	Medium	High
Generic chatbot	Yes	No	Low	Medium
Cloud document chatbot	Yes	Limited	Medium	Low

Table I shows a comparison of common document analysis methods. Manual reading and keyword search keep documents local, so privacy is high. But they do not understand meaning and do not provide citation proof. Because of this, reliability stays at a medium level.

The chatbot systems that are generic comprehend language. They are able to react in a natural manner. Yet they never verify answers to the uploaded document. This results into low reliability because of hallucinated outputs [4], [9].

Document chatbots on clouds provide options of retrieval. Citation support is usually however restricted. Privacy is also undermined as embeddings and data can be stored offsite.

This comparison is a clear indication of the necessity of a superior system. There is need to have a system that can help in semantic retrieval, citation clarity, response accuracy and privacy simultaneously.

D. Disadvantages of Current System

The primary issues of the existing document analysis systems are:

1) There is no profound semantic search in key word search tools [5].
2) Large documents are more time consuming and difficult to read manually.
3) Generative chatbot answers are hallucinatory because they are not grounded on documents [4], [9].
4) Lack of a proper citation-based verification mechanism with a lot of systems.
5) Lacks of confidence scoring or similarity transparency to do validation by user.
6) Loosely integrated systems have poor multi-document ingestion and joint retrieval [3].
7) In case of embeddings and document data storage on third-party servers or in the cloud, privacy is at risk.

8) Vector search systems like FAISS [12] permit indexing on a large scale and similarity search, but the deployment configurations can affect data privacy.
9) Problems with finding specific contextual information within long or complicated documents

E. *Proposed System Motivation*

Due to these shortcomings, a document-based conversational system is required to be structured. The CiteChat model suggested is an extension of embedding-based semantic retrieval [6], dense retrieval systems [5], retrieval-augmented answer generation [4], citation-based display of evidence, and local storage of embedding. Such a design enhances privacy, reliability and transparency.

The system minimizes the false responses by basing responses on the retrieved sections of the document and incorporating the confidence score. It also makes users have confidence in the results in the process of document analysis.

The following section of the review is devoted to related work in retrieval-augmented conversational systems to understand the current research field better and find out the gaps in it.

IV. Related Work

Retrieval-Augmented Generation (also referred to as RAG) is now a significant approach to factual accuracy in large language model systems[4], [15]. It is a combination of document retrieval and answer generation. Connecting external sources of knowledge with generative models, RAG is used to create document-based chat systems that provide answers that are more accurate, clear and easily verifiable.

The question-answering system, which was introduced by Suryavanshi et al. [1], was a Retrieval-Augmented Generation (RAG), which implemented document indexing and embedding-based retrieval models with generative response models. Their work showed a better contextual relevance and less unsupported answers in document-driven application. Pereira et al. [2] suggested Visconde a multi-document question answering system which uses neural reranking methods. Their results revealed that reranking contributes widely to passage relevance which directly affects the final answer quality in RAG pipelines. The paper by Sachan et al. [3] investigated end-to-end training of multi-document retriever and reader models. They placed emphasis on the problems of co-optimization of retrieval and reading component as a way of improving coordination of large document collections.Retrieval- Augmented Generation as a single framework that integrates neural retrieval and sequence-to-sequence generation was formally introduced by Lewis et al. [4]. This ground breaking work formed the theory of the current RAG systems. Karpukhin et al. [5] suggested the Dense Passage Retrieval (DPR) using dual-encoder-based systems to obtain the relevant passages within large-scale corpora and dramatically enhancing the performance of open-domain question answering. Sentence-BERT, which was proposed by Reimers and Gurevych [6], is a technique that can create semantically meaningful sentence embeddings, facilitate similarity search and enhance document retrieval systems.

Fan et al. [7] introduced a survey of retrieval-augmented language models, covering the different architecture designs, retrieval techniques and current gaps in research. Arslan et al. [8] also reviewed the RAG-based systems and explained some practical issues including the methods of hallucination mitigation and optimization of retrieval. Hu et al. [9] suggested a hallucination detection model, LRP4RAG, which uses the relevance propagation methods to detect unsupported detected content in RAG systems.Guu et al. [10] proposed REALM, a retrieval-augmented language model, which means that document retrieval is incorporated in the pretraining phase, and the model can dynamically search external knowledge. Devlin et al. [11] introduced BERT, a bio-directional transformer model, on which most embedding-based retrieval algorithms in RAG systems are based. Johnson et al. [12] proposed FAISS, a very efficient similarity search library aimed at indexing large vectors and it has become an essential part of modern embedding retrieval pipelines.

In general, previous studies indicate that retrieval-enhanced systems enhance accuracy, reasoning and transparency of chat systems. Advances in dense retrieval, semantic embeddings, reranking strategies, hallucination detection and multi-document coordination approaches have made document-based conversational systems much stronger[5], [6], [9], [14]. Still, some issues remain. Confidence scoring, coordination of multiple documents, and reliable answers filtering should be paid more attention. The CiteChat system proposed is based on these concepts. It extends the citation display, similarity-based confidence scoring, hallucination control and local embedding storage to enhance reliability and privacy.

Table II compares traditional chatbots, systems based on RAG, and CiteChat. This table underscores the enhancement of transparency, reliability checks and document-grounded responses by CiteChat.

TABLE II. Comparison Between Existing Chatbots and CiteChat

Feature	Traditional Chatbot	RAG-Based Chatbot	Proposed CiteChat
Uses External Document Retrieval	Not Supported	Supported (Lewis et al. [4], Karpukhin et al. [5])	Supported
Shows Explicit Source Citation	Not Supported	Limited / Implementation-Dependent (Fan et al. [7])	Explicit Citation with Page & Chunk Reference

Feature	Traditional Chatbot	RAG-Based Chatbot	Proposed CiteChat
Confidence Score Estimation	Not Supported	Rarely Implemented (Fan et al. [7])	Similarity-Based Confidence Scoring
Multi-Document Support	Limited Context Handling	Supported but coordination challenges (Sachan et al. [3])	Supports Multi-Document Ingestion and Retrieval
Dense Vector Embedding Retrieval	Not Supported	Supported (Reimers & Gurevych [6], Karpukhin et al. [5])	Sentence-Embedding Based Retrieval
Hallucination Mitigation	Not Supported	Partially Addressed (Hu et al. [9])	Strict Document-Grounded Filtering
Retrieval Re-ranking	Not Supported	Supported (Pereira et al. [2])	MMR-Based Re-ranking
Privacy-Preserving Deployment	Not Supported	Rarely Addressed	Local Embedding Storage

Table II presents a comparison of three types of systems. Traditional chatbots do not use document retrieval mechanisms or use pretrained knowledge to ensure the transparency and fact verification. Conversely, retrieval-augmented generation (RAG) systems combine dense retrieval procedures, including DPR [5], embedding models, such as Sentence-BERT [6], and indexing systems, such as FAISS [12]. The methods contribute to contextualization and factual consistency. Nevertheless, display and estimation of confidence are not always enforced across RAG systems [7]. Such re-ranking strategies [2] and hallucination detection mechanisms are other research studies that have enhanced reliability of the system. In spite of these developments, there are still problems of successfully managing various documents [3] and privacy preserving deployment in practical settings.

CiteChat further builds upon these models of RAG by adding dense semantic retrieval, local FAISS storage, explicit citation with page number, confidence scoring based on similarity, MMR reranking, and document based filtering. This hybrid arrangement is aimed at pragmatically oriented needs of transparency, reliability, and privacy in document based conversational assistants.

V. Proposed System

A. System Overview

CiteChat is a chat assistant that operates on a document basis. It scans the PDF files uploaded by the user and provides evidence-based responses. The system is based on Retrieval-Augmented Generation model [4].

CiteChat, unlike regular chatbots, relies on dense vector embeddings in document chunks by search, as opposed to trained knowledge [5], [6]. The architecture is illustrated in Fig. 1.

Fig. 1. General design of the proposed CiteChat system.

The system comprises of five major sections:

1) Preprocessing and document ingestion.
2) Embedding creation
3) Local vector storage
4) Similarity-based retrieval
5) Generation of answers based on the context.

B. System Architecture Description

The architecture provides a description of the system.

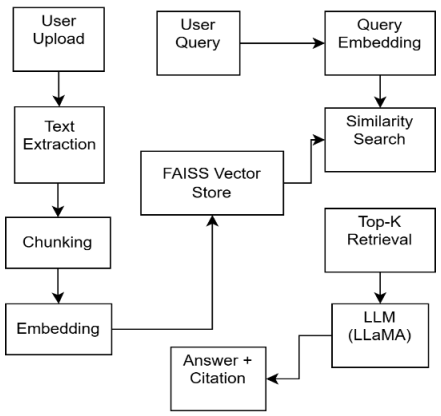

Fig. 2. Internal system architecture of CiteChat showing document processing, embedding creation, vector storage, retrieval, and LLM-based generation.

Fig. 2 illustrates the internal design of CiteChat. It describes the interaction between document processing, and vector storage and retrieval and language model.

Generation can be represented as:

$$e = f(t) \quad (3)$$

In this case, t represents the text chunk and f(.) represents the embedding function. Local storage eliminates transmission of document data to external servers.

C. Pipeline Document Processing

1) Text Extraction The system reads PDF documents using Python libraries.
2) Chunking: The text is broken down into set pieces.
3) Generation Embedding: Every segment is a semantic vector.

D. Retrieval Mechanism

The query converts to an embedding as well when a user issues a query. Similarity between query vector q and each

document vector di is measured with the help of cosine similarity in the system:

$$S_i = cos(e_q, e_{d_i}) = {}^{e_q \cdot e_{d_i}}\Big/_{||e_q||\,||e_{d_i}||} \quad (4)$$

e_q = embedding of the query e_{d_i} = embedding vector of the i-th document chunk. $||\cdot||$ = Euclidean norm (vector magnitude)

The system picks the best-k document fragments with the highest scores of similarity. Maximal Marginal Relevance (MMR)-based re-ranking is used with initial cosine similarity ranking in order to enhance diversity of retrieved segments and minimize redundancy.

Algorithms 1: The Similarity-Based Retrieval in CiteChat.

Input: Query q, Vector store V

Output: Retrieved document set

1) Emb query: q = Emb(q)
2) Calculate cos(e_q, d_i) = compute similarity score.
3) Arrange the pieces of the documents in a descending order in terms of S_i.
4) Pick the k best scoring pieces.
5) The re-ranking based on MMR is used after the cosine similarity ranking [2].
6) Return refined document set D_r.

E. Response Generation Module

The system is taken through the RAG process [4] after retrieval. The language model is conditional on the query posed by the user and the retrieved set of documents in order to produce the final response.

$$R = LLM(q, Dr) \quad (5)$$

In this case, q is query and Dr is the set of documents that are retrieved. LLM is the language model. Fig. 3 illustrates such a workflow.

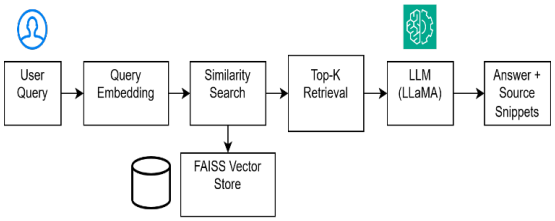

Fig. 3. Query processing pipeline that depicts query embedding, similarity search, retrieving the relevant chunk and context conditioned response generation.

F. Support of Evidence and Estimation of Confidence

A similar case is demonstrated in CiteChat, as well. It displays:

1) Source snippets
2) Page numbers
3) Chunk identifiers

This assists the users to check the answer with ease.

To estimate confidence similarity scores are used:

$$Conf = \frac{1}{k}\sum_{i=1}^{k} cos(e_q, d_i) \quad (6)$$

In this case, k is the number of document chunks ranked highest retrieved to generate an answer. In case the similarity is too low, the system gives out:

"Not mentioned in the given document."

This prevents unsubstantiated outputs. Retrieval augmented systems control hallucination [9].

G. Workflow Explanation

The full workflow is shown in Fig. 4

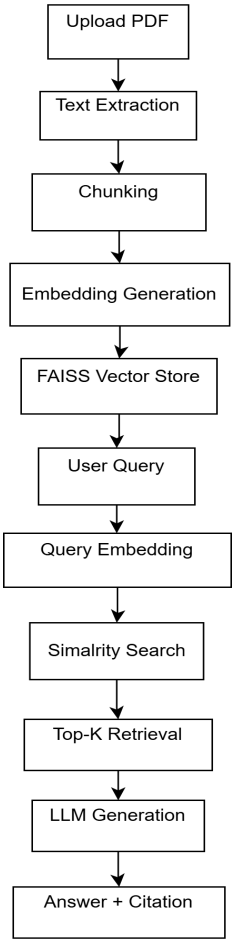

Fig. 4. End-to-end operational workflow of the proposed CiteChat document-grounded chatbot system

H. Key Contribution of Proposed System

The key contributions of the proposed CiteChat system are highlighted below:

1) *Architecture Unified Reliability-Oriented RAG Architecture.*

In contrast to regular RAG pipelines, which are mainly concerned with the accuracy of retrieval [4], [5] CiteChat combines:

a) Dense semantic retrieval [5], [6]
b) MMR-based re-ranking
c) Page level/Chunk level referencing Explicit citation tracing
d) Similarity-based confidence calibration.
e) Extreme document-based response filtering.
f) Reduction of the hallucinatory rate using a strict similarity-based document filtering.

This builds a reliability-driven RAG pipeline whereby generation of answers is limited by retrieved semantic evidence and with verifiable evidence.

2) *Confidence Estimation with Calibration by Similarity*

The majority of currently existing RAG systems do not include quantitative confidence scores [9]. In CiteChat, there is a confidence measure based on similarity:

$$Conf = \frac{1}{k} \sum_{i=1}^{k} cos(e_q, d_i) \quad (9)$$

e_q , d_i are the query and document embeddings respectively.

The retrieval similarity is converted to a measurable indicator of reliability, and the evaluation of the answers can be interpreted and made transparent by this mechanism.

3) Local vector Indexing - Privacy

Even though most RAG systems are based on the framework of the vector indexing system, like that of FAISS [12], external storage services can be deployed. CiteChat locally generates embeddings and FAISS indexing.

This ensures:

a) None of the transmission of documents to third parties.
b) Reduced privacy risks

The system is deployment-focused, which will qualify it in the academic, HR, and enterprise applications.

4) Multi-modality Integrated Document Context Handling

CiteChat can support scalable multi-document ingestion and retrieval capability in a lightweight

architecture whereas single-document and multi-document QA systems treat these two concepts distinctly and do not allow unified multi- PDF ingestion into the system [3].

5) *Experimentation using Controlled Baselines*

The research gives systematic analysis through:

a) Retrieval precision analysis
b) Measurement of accuracy of responding.
c) Citation coverage.
d) Validation of confidence consistency.
e) Comparison of baseline on 675 queries.

The findings support the fact that structured retrieval conditioning is very effective in enhancing factual reliability in document based conversational systems.

I. Performance Evaluation Summary

System performance was tested using retrieval precision, answer accuracy, Citation Coverage, and confidence score.

TABLE III. CONFUSION MATRIX

	Predicted Correct	**Predicted Incorrect**
Actual Correct	TP	FN
Actual Incorrect	FP	TN

Accuracy was calculated using this below formula equation(7):

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (7)$$

Hallucination means answer, which contains information not supported by retrieved document chunks

So metric:

$$HR = \frac{\text{Number of unsupported Answers}}{\text{Total Answers}} \quad (8)$$

Where, HR represents Hallucination Rate.

VI. RESULTS

A. Experimental Setup

The assessment of the CiteChat was conducted with the help of a set of 45 user-posted PDF documents. These covered the resumes, academic lecture notes and research papers. Each document was set with 15 test queries to evaluate retrieval and generation of the answers. Thus, 675 queries were considered in total.

Where N_d is the number of uploaded PDF documents and N_q is the number of queries. In this setup:

$$N_d = 45$$
$$N_q = 15 \text{ x } N_d = 675$$

Preprocessing was done where the individual documents were divided into fixed-length chunks. The number of tokens in each chunk was about 800 with a small overlap to neighboring chunks so as to maintain a continuity of context. This size was selected to retain sufficient meaning and at the same time to make a correct retrieval. Where L_c is the average length of the chunks:

$$L_c \approx 800$$

Sentence Transformer model was used to perform semantic embeddings. One storing of each embedding was a D-dimensional space, in which:

$$D = 384$$

To retrieve, document chunks in terms of their meaning relative to the query were ranked using cosine similarity. The k most relevant chunks were picked and inputted into the language model to generate answers. In this experiment:

$$k = 4$$

The experiments were carried out on a system with an Intel i5 processor, RAM of 16GB and CPU only setting. Embedding generation and similarity search were run locally with the help of FAISS and Sentence Transformer, whereas the language model inference was done with the help of the Groq API (Llama 3.3 70B model).

The Python 3.10 operating system with the FAISS (faiss-cpu) and Sentence Transformer as well as Streamlit, LangChain, NumPy, PyPDF, and python-dotenv were mandatory requirements.

B. System Performance Results

A number of measures were used to test CiteChat. These were retrieval precision, accuracy of response, coverage of citation, consistency in the rating of confidence score, and response time.

All the answers obtained were manually verified with the original papers. The number of the relevant chunks of the retrieved results was computed as retrieval precision. Response accuracy was a measure of the number of responses that were supported by the content of documents.

The results are shown in Table IV.

TABLE IV. CITECHAT SYSTEM PERFORMANCE

Evaluation Parameter	**Observed Value**
Retrieval Precision	0.90
Response Accuracy	0.93
Citation Coverage	0.95
Confidence Score Consistency	0.91
Average Response Time	1.8 sec

As shown in Table IV, the system achieved a retrieval precision of 0.90 and a response accuracy of 0.93. The citation coverage of 0.95 means that most answers included clear document evidence. Confidence score consistency of 0.91 shows that similarity-based scoring matched well with the actual strength of retrieved content.

When compared to keyword search and normal chatbots, these results show clear improvement in document-grounded answering. The gains mainly come from embedding-based semantic retrieval [5] and context-based generation using the RAG model [4].

The average response time was 1.8 seconds per query. This shows that using document grounding and local embedding storage does not slow down the system much.

Therefore, the system is suitable for real-time document interaction.

C. Hallucination Rate Analysis

To evaluate the effectiveness of strict document-grounded filtering, we measured the hallucination rate before and after applying similarity-based confidence filtering.

Hallucination Rate is defined as:

$$HR = \frac{N_{unsupported}}{N_{total}} \quad (10)$$

TABLE V. HALLUCINATION RATE CONTROL

Configuration	Hallucination Rate
Without Filtering	0.19
With Filtering (CiteChat)	0.06

To quantify the improvement achieved through similarity-based filtering, the hallucination reduction percentage is calculated as:

$$Reduction = \frac{HR_{before} - HR_{after}}{HR_{before}} \times 100 \quad (11)$$

The proposed similarity-based filtering mechanism reduces hallucinated responses by approximately 68.4%, demonstrating significant improvement in factual reliability and document-grounded answer generation.

D. Precision Accuracy Relationship Analysis

Fig. 5 shows how retrieval precision and response accuracy are related across different document QA systems. The results show a clear positive relationship between retrieval precision and response accuracy. Systems with low retrieval precision, such as keyword search, had lower answer correctness. This is because they often retrieved incomplete or unrelated text. In contrast, CiteChat achieved better answer quality by selecting only the most meaningful document chunks. By limiting generation to the top k relevant segments, the system reduced unsupported responses and improved factual consistency. This confirms that retrieval quality directly affects answer reliability in retrieval-based systems. Better retrieval leads to better generation.

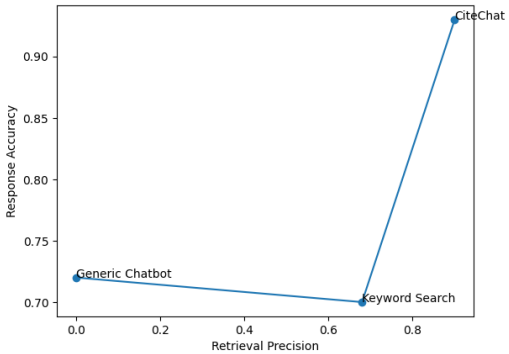

Fig. 5. Relationship Between Retrieval Precision and Response Accuracy

A comparative evaluation was conducted between CiteChat and baseline systems across the evaluated queries. The observed improvement in response accuracy was consistent across documents, indicating stable performance gains.

E. Impact of Query Reformulation

TABLE VI. QUERY REFORMULATION COMPARISON

Configuration	Retrieval Precision	Response Accuracy
Without Reformulation	0.87	0.90
With Reformulation	0.90	0.93

TABLE VI shows that query reformulation improved semantic alignment and resulted in measurable gains in retrieval precision and response accuracy.

TABLE VII. BASELINE TABLE

System	Retrieval Precision	Response Accuracy
Keyword Search	0.68	0.70
Generic Chatbot (No Retrieval)	0.00	0.72
CiteChat (Proposed)	0.90	0.93

Table VI demonstrates the relative effectiveness of the proposed system of CiteChat with baseline strategies. The search of key words attained low precision of retrieval as there was no semantic understanding. The generic chatbot, which is not based on document retrieval, was less accurate in responses as the answers were created without the grounding of documents. Comparatively, CiteChat achieved much greater retrieval precision and accuracy in response generation due to the combination of dense semantic retrieval and retrieval-augmented generation. This analogy validates the fact that structured retrieval conditioning enhances the factual reliability and response quality on the basis of documents.

F. Ablation Study

TABLE VIII. ABLATION STUDY

Configuration Accuracy	Retrieval Precision	Response Accuracy
Without MMR	0.86	0.89
Without Reformulation	0.87	0.90
Absence of Confidence Filtering	0.88	0.91
Full CiteChat Model	0.90	0.93

Table VIII is an assessment of the contribution of all significant elements in the proposed architecture. The

elimination of MMR-based re-ranking eliminated retrieval precision as a result of more redundancy in the selected chunks. Omitting the reformulation of queries reduced semantic correspondence between queries and document content. Likewise, the inability to use the confidence based filtering minimally lowered the reliability of the responses. The overall performance of the full CiteChat model was the best, which confirms the presence of contribution to the quality of document-grounded responses by every module.

G. Error Analysis

Although the overall performance was good, few limitations were noticed in the course of experiment assessment.

1) Reformulation Misalignment

In some instances query reformulation with the use of LLM offered alternative words that were semantically accurate but not directly found in the document. This at times led to a very slight decrease in the cosine similarity scores, retrieval precision of domain specific words.

2) Semantic Incompleteness:

Even though cosine similarity is efficient in determining relevant chunks, high similarity does not necessarily imply that the retrieved passage includes enough contextual information to be able to answer the query to the fullest. In some cases, the retrieved chunks were topically relevant, but they did not have all the information supporting them.

On the whole, these restrictions were seen to occur in a minimal percentage of instances and did not have a significant impact on the reliability of the system. To further alleviate such problems, future work can investigate strategies of adaptive chunking and semantic-constrained reformulation control.

VII. Discussion

The findings indicate that retrieval-based conditioning enhances consistency and sharpness of document chat systems. Normal language model chatbots create answers based on what they have learnt through training only. CiteChat, in contrast, constrains the generation of answers by first picking meaningful pieces of documents. It is then this retrieved context which the model uses in response. Due to this fact, the number of unsupported answers is minimized and the consistency of facts becomes better.

The increased accuracy of the retrieval in the assessment indicates that dense semantic embeddings can retrieve meaning when there is no exact matches between words. This confirms previous research that points out that embedding-based recall is better suited to semantic tasks compared to the simple key word search [5]. In the case where the system only allows generation of the top k of the relevant chunks, the risk of hallucinated or irrelevant answers is significantly reduced. Citation support is also another powerful attribute. With the system, the snippets of the source and the page numbers and the chunk IDs are displayed with every answer. This will enable the users to verify the document on their own. Another great contribution of this system is privacy. The entire embeddings and indexing of vectors is local. The outside servers are not transmitting sensitive documents.

In contrast to the existing systems on RAG, which merely concentrate on retrieval accuracy, CiteChat incorporates a calibrated confidence estimation and hard document-based response filtering into a single privacy-preserving system.

VIII. Conclusion and Future Work

The current paper introduced CiteChat, a document-based conversational system that was developed in the framework of a Retrieval-Augmented Generation model to enhance reliability in document question answering tasks. The given design is a combination of the dense semantic retrieval, similarity-based document selection, context-based response generation, citation, and confidence scoring in a structured pipeline. The experimental outcomes demonstrated that the factual accuracy and the decrease of unsupported responses are higher when providing grounding of the language model on retrieved document content, in contrast to the traditional chatbot systems. To make the process more transparent and safeguard user privacy, the clear citation display and the storage of local embedding are added.

Although the system has had good retrieval precision and high response accuracy, there are still few limitations. The assessment was done based on a domain-oriented dataset, and the type of documents was not very varied. The performance can be different in case of testing it on bigger and more complicated document collections. The scalability with regards to retrieval is also another issue with more document chunks stored.

The future direction of work will consist of the integration of both semantic and key-word based retrieval in a hybrid configuration.

References

[1] K. Suryavanshi, N. Thikekar, R. Pawar, and S. Ashtekar, "Implementation of RAG Based Question-Answering Application,"
in Proc. 2025 Int. Conf. Data Science and Business Systems (ICDSBS),
Chennai, India, 2025, pp. 1–6, doi: 10.1109/ICDSBS63635.2025.11031968.

[2] J. Pereira, R. Fidalgo, R. Lotufo, and R. Nogueira,
"Visconde: Multi-document QA with GPT-3 and Neural Reranking,"
arXiv preprint arXiv:2212.09656, 2022.

[3] D. S. Sachan et al.,
"End-to-end training of multi-document reader and retriever for open-domain question answering,"
arXiv preprint arXiv:2106.05346, 2021.

[4] P. Lewis et al.,
"Retrieval-augmented generation for knowledge intensive NLP tasks,"
arXiv preprint arXiv:2005.11401, 2020.

[5] V. Karpukhin et al.,
"Dense passage retrieval for open-domain question answering,"
in Proc. EMNLP, 2020, pp. 6769–6781.
[6] N. Reimers and I. Gurevych,
"Sentence-BERT: Sentence embeddings using Siamese BERT networks,"
arXiv preprint arXiv:1908.10084, 2019.
[7] W. Fan et al.,
"A survey on retrieval-augmented large language models,"
arXiv preprint arXiv:2405.06211, 2024.
[8] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz,
"A survey on retrieval-augmented generation with large language models,"
Procedia Computer Science, vol. 236, pp. 1571–1580, 2024.
[9] H. Hu, C. He, X. Xie, and Q. Zhang,
"LRP4RAG: Detecting hallucinations in retrieval-augmented generation,"
arXiv preprint arXiv:2408.15533, 2024.
[10] K. Guu et al.,
"REALM: Retrieval-augmented language model pre-training,"
arXiv preprint arXiv:2002.08909, 2020.
[11] J. Devlin et al.,
"BERT: Pre-training of deep bidirectional transformers for language understanding,"
in Proc. NAACL-HLT, 2019.
[12] J. Johnson, M. Douze, and H. Jégou,
"Billion-scale similarity search with GPUs,"
arXiv preprint arXiv:1702.08734, 2017.
[13] T. B. Brown et al.,
"Language models are few-shot learners,"
in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.
[14] O. Khattab and M. Zaharia,
"ColBERT: Efficient and effective passage search via contextualized late interaction over BERT,"
in Proc. ACM SIGIR Int. Conf. Res. Develop. Inf. Retrieval, 2020.
[15] J. Gao et al.,
"Retrieval-augmented generation for large language models: A survey,"
arXiv preprint arXiv:2312.10997, 2023.