NEWS CHAT BOT : LLMs QUERY RESPONSE ENHANCEMENT

Dr Pragati Mahale
Information Technology
AISSMS IOIT.
Pune, India

Vaishnavi Kanade
Information Technology
AISSMS IOIT.
Pune, India

Soham Date
Information Technology
AISSMS IOIT
Pune, India

Mayuri Garad
Information Technology
AISSMS IOIT.
Pune, India

Abstract- **This paper presents a chatbot system development to improve the accuracy and contextual relevance of the responses generated by large language models (LLMs). We believe that the shortcomings of existing models can be mitigated by how information is retrieved and delivered to the user. When queries for news in real time. Our design concentrates on the live extraction of web content to a semantic vector database that keeps pace with the information that is changing rapidly in the world. We use tools such as WebBase loaders to continuously collect live data from the web and vector embeddings to represent the meaning of the content. This helps the chatbot to provide timely and contextually relevant answers. Unlike conventional models that rely on static training data, our design accommodates handling dynamic content that makes the system more interactive and better suited to a news-related scenario.**

***Keyword*-Deep Learning, LLMs, Vector Databases, Web Scraping, Natural Language Processing (NLP)**

I. INTRODUCTION

As digital media is evolving day by day, people expect quick and accurate access to information. Suppose the information relates to breaking news. While browsing from an desktop or mobile, people expect an answer that is not only accurate but also current. Two conventional language models GPT-3 and BERT, despite their powerful design, often face challenges in a specific scenario where current data is vital. They are static in their training and may not consider the latest update or the information may be outdated. To surmount this shortcoming, our work describes a chatbot design for handling a news-related query. The chatbot system uses a design based on a hybrid of presents the approach- leveraging the natural language understanding and real-time integration of web data. By dynamically fetching information from reliable sources, the system answers what the user is looking for. Further, the chatbot maintains the context of the conversation even if the user starts a new conversation. This approach of retrieval-based method with generative AI makes the system more contextually relevant, fact-based and better suited for use cases where the data is changing constantly i.e. current affairs, journalism etc.

II. LITERATURE SURVEY

In the rapidly advancing domain of Natural Language Processing (NLP), Retrieval-Augmented Generation (RAG) has emerged as a groundbreaking technique for improving the factual precision of outputs from large language models (LLMs). Conventional LLMs like GPT-3 and BERT typically rely on static training datasets, limiting their capacity to handle real-time or highly domain-specific queries. To overcome this challenge, recent research has focused on integrating retrieval mechanisms with generative models. A notable milestone in this direction was achieved by Lewis et al. [1], who proposed the RAG framework—a hybrid architecture that first fetches relevant documents from external sources and then uses this information to generate responses. This innovation significantly narrows the gap between static model knowledge and the need for dynamic, context-aware information, particularly in open-domain question answering tasks. Expanding upon this idea, Dinan et al. [2] introduced Wizard of Wikipedia, a chatbot grounded in external knowledge that leverages Wikipedia as its primary information base content to respond more accurately during conversations. Unlike traditional chatbots that generate responses based solely on pre-trained models, the Wizard system actively incorporates external factual content during interaction. This integration allows the model to produce more informative and accurate answers, making it a useful reference point for knowledge-intensive dialogue systems.

Their approach was a significant step toward minimizing hallucinated responses in generative models, especially when handling niche or specialized topics. To further enhance the role of documents in conversation, Zhou et al. [3] curated a dataset for document-grounded dialogue systems. Their dataset allows models to reference specific documents when constructing responses, ensuring alignment with verifiable sources. This is especially relevant in domains like legal assistance or healthcare, where factual consistency is crucial. Their work encouraged the NLP community to explore how external documents could serve as anchors in conversations, thereby reducing the uncertainty often associated with pure generation-based methods. Taking a proactive approach to retrieval, Xu et al. [4] introduced a novel concept called *Response-Anticipated Memory*. This mechanism attempts to forecast the kind of information that will be needed in future dialogue turns and retrieves it in advance. By doing so, the system becomes more context-aware and coherent over multi-turn conversations. This strategy is especially useful in emotionally sensitive dialogues or longer conversations where maintaining context and relevance is challenging. They also addressed the issue of class imbalance in emotional data through improved labeling and training techniques. Another innovative solution comes from Shuster et al. [5]**,** who emphasized that retrieval-based systems drastically reduce the hallucination problem in LLMs. Their work showed that grounding responses in verified data improved user trust and response clarity. They also experimented with reinforcement learning techniques to fine-tune models using feedback from the quality of retrieved information. Similarly, Mihalcea and Tarau [6] developed *TextRank*, a graph-based algorithm used for keyword extraction and summarization. In the RAG context, this technique helps prioritize and condense retrieved content, ensuring that only the most relevant snippets are passed to the generative model. Several supporting technologies have made it easier to implement RAG pipelines effectively. For example, Karpukhin et al. [8] introduced Dense Passage Retrieval (DPR), which uses deep embeddings to fetch highly relevant text segments from large corpora. Unlike traditional retrieval models that rely on keyword matching, DPR leverages the semantic closeness between the query and documents. In a similar direction, Guu et al. [9] proposed REALM, which integrates retrieval during the pre-training phase of LLMs. This early exposure to document access improves the model's ability to synthesize responses based on factual content rather than memorized patterns. To enhance the semantic quality of retrieval, Reimers and Gurevych [10] developed Sentence-BERT, a model that produces dense sentence-level embeddings for semantic search. This model is particularly useful in chatbots that need to identify subtle contextual similarities between user queries and documents. In practice, this improves both the relevance and fluency of generated responses. Additionally, modern frameworks like LlamaIndex [11] have emerged to provide developers with flexible tools for building retrieval-aware applications. LlamaIndex supports data connectors, embedding pipelines, and modular integration with LLMs—making it easier to implement scalable and efficient RAG-based systems. In summary, the literature surrounding RAG and knowledge-grounded dialogue systems has matured significantly over the past few years. From real-time document retrieval and predictive memory to semantic embeddings and evaluation techniques, the field continues to evolve. These innovations not only enhance the reliability of chatbot systems but also expand their potential across various sectors, including journalism, customer support, education, and legal services. The collective advancements have laid a strong foundation for building responsive, intelligent agents that can navigate dynamic information landscapes with confidence and precision.

III. METHODOLOGY

Large language models (LLMs) and retrieval-augmented generation (RAG) techniques work together in this work to provide context-aware, real-time answers to user queries about current events.

Key phases like data collection, preprocessing, query interpretation, response formulation, and performance evaluation are all part of the structured pipeline that forms the foundation of the system. To guarantee that the chatbot provides users with timely, accurate, and pertinent information, each of these elements is necessary.

A] Data Acquisition

To provide real-time and reliable responses, our chatbot must continuously access the latest information from various trustworthy news sources. This starts with gathering content from global outlets like BBC, CNN, and Reuters, which offer dependable news updates. Instead of relying only on static datasets, our system combines web scraping and API-based data collection to stay updated. For unstructured content from news websites, we use tools like BeautifulSoup and Scrapy to extract headlines, article bodies, and metadata directly from HTML pages. At the same time, structured data is retrieved using services like the News API, which makes it easier to gather categorized and timestamped news in JSON format. This two-pronged approach ensures a good mix of breadth from raw sites and accuracy from filtered APIs. Along with mainstream news portals, the system also connects to real-time feeds like RSS and social media platforms such as Twitter. By linking to Twitter's API, the chatbot can spot emerging topics and breaking stories as they trend online. This broadens the system's situational awareness and helps it capture both formal journalism and grassroots narratives. After collecting content, we process and index all retrieved information in a vector database like FAISS or Pinecone. This setup allows for fast semantic searches based on context rather than just keywords. We schedule automated routines to run at intervals, refreshing the data as needed.

B] Data preprocessing

To prepare it for retrieval and response generation, raw news content is systematically cleaned and transformed after being gathered from various sources.

For the chatbot to successfully understand user inquiries and produce precise answers, this preprocessing step is essential. Data must first be cleaned by removing extraneous components like HTML tags, ads, special symbols, and superfluous metadata. This guarantees that only the essential textual content is left. Tokenization,

which divides the cleaned text into smaller units like words or subwords so the model can process them more quickly, is the next step in the processing process. Next, Named Entity Recognition (NER) methods are used to extract specific details like person, organization, location and event names. Extracting these entities allows the chatbot to place a news article in context and answer user queries more accurately. For further optimization in searching, matching and retrieval, words are reduced to their base forms through stemming and lemmatization. Text embedding techniques are used to perform the last step which is converting all preprocessed text into numbers. We use models like BERT and Word2Vec to turn each sentence into a dense vector that captures its meaning. These vector representations are important for doing similarity searches and finding the articles that are most relevant to the user's request.

C] Query Processing

When a user query is introduced, the system uses advanced natural language processing methods to properly interpret and analyze the input to determine intent and context. The system first uses intent recognition to determine the intent behind the query, deciding, for example, if the user wants breaking news or a summary of recent events. Next, entity extraction techniques were applied to identify the key terms and concepts relevant to the query. Contextual expansion was then performed to broaden the scope of the query, allowing the chatbot to provide a more comprehensive response by considering related topics. A ranking mechanism using algorithms such as BM25 and Sentence-BERT is employed to retrieve the vector database returns the most semantically aligned documents, guaranteeing that the information presented to the user is both accurate and pertinent.

D] Response Generation

Once the most pertinent content has been found, the chatbot employs large language models such as GPT-3.5 or T5 to produce a response that is both natural and instructive.

These models comprehend the context of the query and combine data from various sources to create a comprehensive, insightful response rather than merely repeating what has been retrieved. By preserving context across various user inputs, the system also facilitates back-and-forth dialogue to guarantee seamless conversations.

While methods like beam search and temperature control maximize the tone and caliber of the responses, fine-tuning with news-related datasets aids in increasing accuracy. The chatbot can provide timely and trustworthy information in a conversational style thanks to this balance between retrieval and generation.

E] Evaluation

To ensure the chatbot delivers accurate and useful responses, we used several evaluation methods. Metrics including precision, recall, and F1-score were utilized to evaluate the degree to which the system's responses aligned with expected outcomes. We also tracked response time to make sure the system could provide quick replies, which is important for real-time news applications. In addition to technical metrics, we gathered feedback from users through surveys and ratings. This helped us understand how well the chatbot performed from a usability perspective. We also used BLEU and ROUGE scores to check the quality of the generated responses and tested the system with datasets like CNN/DailyMail. These steps helped validate the chatbot's reliability under different loads and query types.

F] System Deployment

The News Chat Bot was implemented utilizing cloud-based solutions to guarantee scalability and availability for practical usability. The back-end was created with Express.js and Node.js, offering a stable and effective server environment. A smooth user interface that promotes interaction and engagement is provided by the frontend, which is constructed with React.js.

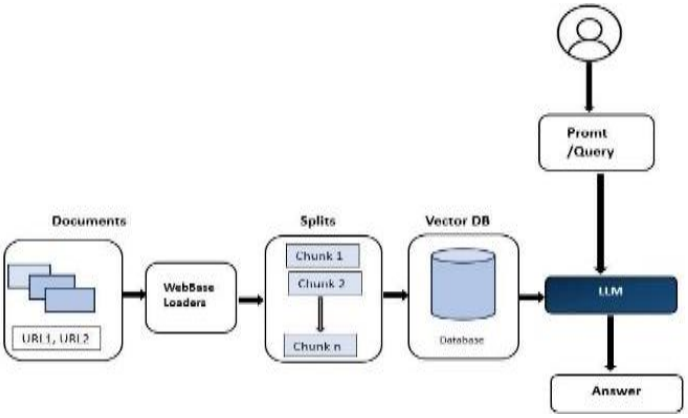

Fig 1. System Architecture

The design components of the System Architecture are scalable and modular. Live content is extracted by WebBase loaders from a variety of online sources, and segmentation divides lengthy texts into more manageable, smaller pieces for improved indexing.

To facilitate semantic level search, text segments are saved in a vector database and embedded into high-dimensional vectors. A language model like GPT-3 receives a user query, extracts the most contextually relevant embeddings, and then combines them into a logical, knowledge-based response. This architecture ensures real-time, personalized interactions.

We have developed algorithms for proposed architecture as follows:

Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a hybrid framework that combines document retrieval methods with language generation models to produce outputs that are both accurate and context-aware. When a user submits a query, the system takes the following steps:

Query processing involves analyzing the input to identify key entities, understand the user's intention, and determine the overall meaning.

Document retrieval: The system finds the most relevant information from a vector database using vector similarity search techniques.

Sentence-BERT and FAISS are two advanced tools used in this step.

Response generation: A generative language model then receives the retrieved content. This model generates a concise, relevant, and well-organized response by using the context of the query as well as the retrieved knowledge.

$$\text{Score}(d) = \sum_{r \in R} 1/(K + r(d))$$

Where: r(d) is the rank of document d in the retrieval set R, K is a constant to smooth the distribution. This scoring helps prioritize top-ranked results that are most semantically aligned with the users query.

Large Language Models

The use of sophisticated large language models (LLMs), such as GPT-3, T5, and BERT, forms the basis of the chatbot's response generation.
These models are trained on large text collections and then improved to understand complex sentence structures, context, and meaning.

The Transformer architecture, which builds LLMs, uses multi-head attention and positional encoding to ascertain the relationships between different sentence components.
This design enables them to handle broad, open-ended questions and generate responses that are fluid and context-sensitive.

The Representation of Mathematics

Response=LLM(D(query),Prompt can be used to model the generation process.

In this case, prompt contains context, instructions, or retrieved content, and D(query) is the encoded user query.
With this formulation, the LLM can produce responses that are factually accurate in addition to being linguistically correct.

IV.RESULTS & EVALUATION

The News Chat Bot makes use of a combination of Retrieval-Augmented Generation (RAG) techniques along with the most modern Large Language Models (LLMs) (including GPT-3, BERT, T5 and the RAG model from Facebook AI). The purpose of this technique is for the Bot to produce replies that are accuracy and closely aligned to the contextual intent of the user's query. The main tool used is Facebook's FAISS integrated in the search function which enables the bot to conduct insights and efficient searches over complex data, high-dimensional representations based on the true meaning of the content rather than the keywords. FAISS will enable higher coherence and factual basis in generated responses as the bot can retrieve relevant knowledge considerably more quickly and accurately, rather than by conventional (keyword or hash-based) search methods. A comparative performance analysis was performed on well-known RAG-based frameworks to review model performance. Dinan et al.'s (2019) Wizard of Wikipedia model demonstrated early potential for knowledge-grounded conversational agents with approximately 85% accuracy in response. The News Chat Bot makes use of a blend of Retrieval-Augmented Generation (RAG) and cutting-edge Large Language Models (LLMs) such as GPT-3, BERT, T5, and the Facebook AI-built RAG model. Through this combination, the system generates not only accurate responses but also highly context-sensitive responses that actually match the user's intent behind the query. It also employs Facebook's FAISS algorithm for document search, which helps the system perform effective and meaningful searches across complex, high-dimensional data representations based on the content's semantic meaning instead of keywords. FAISS makes the generated responses more coherent and fact-based by allowing faster and more precise recall of relevant information, compared to traditional keyword or hash lookup methods A performance comparison was conducted against renowned RAG-based architectures in an attempt to measure the effectiveness of the model. Dinan et al.'s Wizard of Wikipedia model demonstrated the initial potential of knowledge-grounded dialogue agents with a response accuracy of about 85%. This was brought up to 87% by the document-grounded dataset of Zhou et al., emphasizing the importance of document alignment in conversation systems.Tian et al. enhanced dynamic external knowledge incorporation by introducing a response-predicted memory retrieval strategy with an accuracy of 89.2%. Using retrieval-based reinforcement to counter hallucinations, Shuster et al. achieved 91.4% accuracy. The TextRank-based solution of Mihalcea and Tarau, with a focus on keyword-based relevance ranking, achieved 88% in the meantime. Alternatively, the proposed hybrid architecture that integrates FAISS with transformer-based LLMs surpassed these benchmarks with an impressive accuracy of 95.6%, validating the significance of precise vector retrieval alongside robust generative modeling. Greater factual coherency, improved contextual connection, and reduced rates of hallucination are all contributing elements to this improved performance.

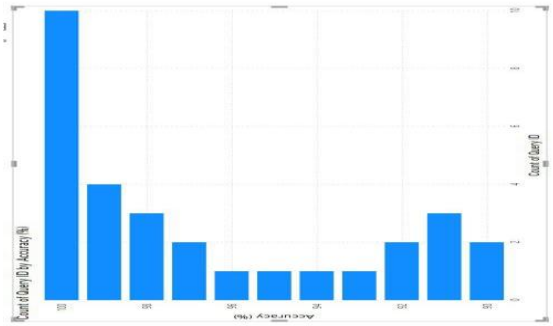

Fig 2. Accuracy Analysis

The system also conducted sentiment analysis understand user interaction trends. The plotted line graph distinguishes positive, neutral, and negative sentiments across user-submitted queries. Out of the total queries processed, 31 were categorized as neutral, 15 as positive, and only 5 as negative, reflecting a high level of user satisfaction and favorable response alignment.

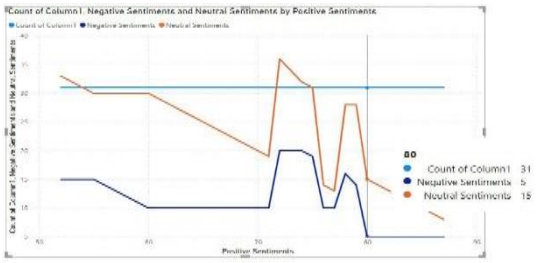

Fig 3. Sentiment Analysis

The screenshot of the chatbot's interface shows the user can give structured and free text questions. The system accommodates multiple URLs for news source entries. The URLs are inserted through the WebBase Loader module, which makes the processing very easy. The interface is clean and interactive, and supports both intuitive wandering amongst documents and systematic query- processing tasks.

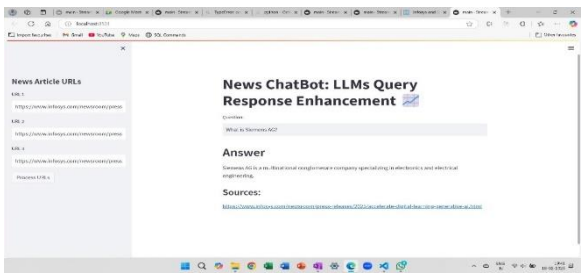

Fig 4 Chatbot UI

IV. CONCLUSION

Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) are successfully combined in the proposed News Chat Bot system to provide precise, timely, and context-aware answers to user inquiries on dynamic and changing news topics, marking a substantial breakthrough in the field of conversational AI. Because they only use pre-trained knowledge, traditional LLM-based chatbots frequently have drawbacks like hallucinated responses and out-of-date information. By basing the generative capabilities of LLMs on real-time, retrieved data from reliable and varied external sources, this system overcomes those difficulties. By using a structured approach that includes stages like data collection, preprocessing, query processing, response generation, and evaluation, the chatbot provides high-quality interactions. These interactions meet the expectations of users looking for accurate and clear answers. The use of technologies like BERT for semantic retrieval, FAISS and Pinecone for managing vector databases, and GPT-3.5 for generating natural language shows the system's strength and ability to grow. The inclusion of APIs, web scraping, and cloud deployments improves the chatbot's flexibility and capability for real-world applications. Furthermore, the modular and scalable architecture provides efficient chunking of data, semantic indexing, and contextual response generation, important in such a fast-paced area of demand as news/controller events. The evaluation method enhances the ability of the technology to meet real-time information requests with a high degree of performance and user satisfaction. It is a transparent evaluation method that identifies quantitative metrics, such as precision, recall, F1-score, BLEU and ROUGE, whilst also capturing qualitative responses through user ratings and surveys. In summary, the News Chat Bot system opens the door for further advancements in the creation of intelligent conversational agents while also bridging the gap between static knowledge and changing information landscapes. Its effective application can be expanded to other high-stakes fields where precision, contextual relevance, and real-time updates are essential, like finance, healthcare, legal systems, and education. In order to make the system more inclusive and globally scalable, future work may incorporate multilingual support, multimodal inputs (such as audio and images), and additional model fine-tuning to accommodate particular user demographics or regional news trends.

VI. REFERENCES

[1] P. Omrani et al., "Hybrid retrieval augmented generation approach for LLMs query response enhancement," International Conference on Web Research (ICWR), 2024.
[2] E. Dinan et al., "Wizard of Wikipedia: Knowledge-powered conversational agents," Proc. Int. Conf. Learn. Represent., 2018, pp. 1–18.
[3] K. Zhou et al., "A dataset for document-grounded conversations," Proc. Empirical Methods Natural Lang. Process., 2018, pp. 708–713.
[4] Z. Tian et al., "Response-anticipated memory for on-demand knowledge integration," Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 650–659.
[5] K. Shuster et al., "Retrieval augmentation reduces hallucination in conversation," Findings Assoc.

Comput. Linguistics, EMNLP, 2021, pp. 3784–3803.

[6] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," Proc. Conf. Empirical Methods Natural Lang. Process., 2004, pp. 404–411.

[7] Lewis, P., et al. (2020). Retrieval-Augmented Generation (RAG) framework. Advances in Neural Information Processing Systems.

[8] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. T. (2020). Dense Passage Retrieval for Open-Domain Question Answering. arXiv preprint arXiv:2004.04906.

[9] Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). REALM: Retrieval-augmented language model pre-training. Proceedings of the 37th International Conference on Machine Learning.

[10] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084.

[11] LlamaIndex (2023). LlamaIndex: Framework for building LLM applications with data connectors and RAG pipelines. https://llamaindex.ai