# Context-Aware Conversational AI: Retrieval-Augmented System with Dynamic Learning from Web Content

Author
Arjun Jagdale, *arjunjagdale.sits.entc@gmail.com*

Instructors
Prof. Makarand Patil, *makarandpatil.sits@sinhgad.edu*
Prof. Anagha Chaphadkar, *anagha.chaphadkar.sits@sinhgad.edu*
Prof. Swanand Pasalkar, *swanandpasalkar@mmcoe.edu.in*

Dept of Electronics and Telecommunication, Sinhgad Institute of Technology and Science, India
Dept of Electronics and Telecommunication, Marathwada Mitra Mandal's College of Engineering, India

## Abstract
Retrieval-Augmented Generation (RAG) has emerged as a powerful approach to improve the contextual understanding of language models by integrating external knowledge. This paper presents a lightweight, offline-capable RAG system that leverages the Mistral 7B Instruct model to deliver context-aware responses using dynamically scraped web content or user-uploaded text files. Unlike conventional chatbot systems that rely solely on pre-trained knowledge, our architecture integrates a custom web scraper, FAISS-based embedding indexer, and persistent memory to continuously learn and adapt to new information in real time. The model accepts user queries, retrieves relevant context from indexed documents or URLs, and generates responses using prompt engineering tailored for instruction-based LLMs. This setup not only improves the relevance of answers but also offers a practical, fully local alternative to cloud-dependent AI services. The system demonstrates strong potential for educational tools, research assistants, and secure enterprise applications.

## 1. Introduction
*What is the problem we are solving?*
Modern conversational AI systems often struggle to provide accurate and up-to-date responses when faced with dynamic or domain-specific content. Most chatbots rely on static, pre-trained knowledge and cannot incorporate new information unless explicitly retrained. This limits their usefulness in scenarios where context is constantly evolving, such as educational resources, personal research, or rapidly updating websites.
*Why is context-aware QA important?*
Context-aware Question Answering (QA) systems enhance the relevance and quality of responses by grounding answers in external, real-time knowledge. By retrieving supporting content from documents or the web, these systems can reason over recent or domain-specific information, making them more accurate and useful. For users engaging in academic research, professional tasks, or learning, having a chatbot that understands the specific context of a topic can significantly improve productivity and understanding.

*Why use local models?*
While cloud-based models like GPT-4 or Claude offer powerful capabilities, they come with challenges — privacy concerns, limited customization, latency, and cost. In contrast, local models such as Mistral 7B Instruct, when paired with efficient tooling like llama-cpp-python, allow for complete offline execution. This approach ensures data security, supports use in low-resource or restricted environments, and allows users to fine-tune the experience for their own needs without external dependencies.

## 2. System Design
The proposed system is designed to function as a retrieval-augmented chatbot that processes user-provided content (via URL or file), builds a semantic index, and enables interactive question

answering using a locally deployed Mistral 7B Instruct model. The architecture is modular, consisting of four main stages:

*2.1 Input: URL or File-Based Source*

Users begin by providing either:

1. A web URL, from which the system scrapes textual content using a custom HTML parser, or
2. A plain text file, loaded directly from local storage.

This flexibility enables dynamic integration of both online and offline sources for contextual grounding.

*2.2 Preprocessing: Web Scraping or File Loader*

1. If a URL is provided, the scrape_url() function in web_scraper.py extracts clean body text from the HTML content using BeautifulSoup, removing navigation bars, scripts, and advertisements.
2. If a file is specified, the load_text_file() function reads and parses the text into manageable chunks for further processing.

This ensures all content is uniformly preprocessed, regardless of the input source.

*2.3 Embedding and Indexing: Semantic Understanding with FAISS*

The core of the system's retrieval capability is built using:

1. Sentence Embedding: Leveraging a transformer-based encoder (e.g., all-MiniLM-L6-v2), the textual data is transformed into high-dimensional semantic vectors that capture meaning.
2. FAISS Indexing: These vectors are stored in a FAISS index for fast approximate nearest neighbor search. This enables the system to retrieve the most relevant context segments in real-time when a question is asked.

This lightweight embedding + indexing step gives the local system its "memory" of the documents/web pages.

*2.4 Response Generation: Mistral 7B Instruct with Custom Prompts*

Once the user poses a question:

The system uses the indexed FAISS store to retrieve top-k relevant chunks based on semantic similarity. These chunks are combined with the question to form a contextual prompt in the [INST] ... [/INST] format, suitable for the Mistral model.

The local LLM (llama_cpp.Llama) processes this prompt and generates a natural language response using quantized inference (GGUF format).

The prompt is carefully structured to guide the model to ground its answers in the retrieved context, rather than relying solely on pre-trained weights.

# 3. Implementation and features

The system has been implemented in Python with an emphasis on modularity, performance, and offline capability. The design leverages open-source libraries for natural language processing, vector similarity search, and large language model inference. Key implementation highlights are detailed below.

*3.1 Local LLM Inference using Mistral 7B*

At the core of the system is the Mistral-7B-Instruct model, a powerful instruction-tuned language model. The quantized .gguf variant of this model is run using the llama-cpp-python backend, which allows efficient CPU or GPU inference on local machines without requiring internet access. Key configurations include:

- n_ctx = 2048 for handling long prompts
- Multi-threading (n_threads = 8) for faster inference
- use_mlock=True to lock the model in RAM for performance

This setup enables near real-time responses while ensuring data privacy and complete control over model execution.

*3.2 Semantic Search via Sentence Embeddings + FAISS*

To enable contextual question answering, the system performs semantic indexing using:

- Sentence Transformers (e.g., all-MiniLM-L6-v2) to convert text into embeddings
- FAISS (Facebook AI Similarity Search) to store and retrieve context based on similarity to the user query.

The result is a highly responsive retrieval system that narrows down relevant information before passing it to the LLM.

*3.3 Persistent Memory with JSON*

The chatbot maintains a persistent memory system using a chat_memory.json file. This file stores:

- Long-term memory or background knowledge (manually editable)
- Full chat history between the user and the LLM.

This memory is automatically loaded and updated during runtime, allowing the chatbot to remember past interactions and adapt to user input over time.

*3.4 Dynamic Prompt Construction*

All responses are generated using a dynamically built prompt in the following format:

```
[INST] Use the following context to answer
the question:

<retrieved context>

Question: <user’s query> [/INST]
```

This ensures the LLM is always grounded in retrieved facts, improving answer accuracy and relevance while preventing hallucinations.

*3.5 Input Flexibility: File or URL*

The user can choose between two modes of feeding data:

- URL Mode: Web content is scraped live using BeautifulSoup and passed to the embedder.
- File Mode: Local .txt files are read directly.

This dual-mode input design allows the system to be used for both real-time knowledge extraction and offline document-based Q&A.

# 4. Results

To evaluate the effectiveness of the developed RAG-based system, we tested it using a publicly available, knowledge-rich webpage: [https://www.britannica.com/animal/zebra.](https://www.britannica.com/animal/zebra)

Upon running the command-line interface with the URL and local Mistral model path, the system initialized within approximately 20 seconds, loading both the content and the model.

```
Scraping from URL: https://www.britannica.com/animal/zebra
Loading Mistral model...
llama_init_from_model: n_ctx_per_seq (2048) < n_ctx_train (32768) -- the full capacity of the model will not b
e utilized

Ask a question (or type 'exit'): Why Do Zebras Have Stripes? Explain in brief.
```

Click [here ](#)for broader view of images

*Example Questions and Responses:*

*Q1: Why Do Zebras Have Stripes?*

Mistral: “Zebras have stripes to help them blend in with their surroundings and avoid detection by predators. The stripes make it difficult for predators to single out an individual zebra from a group. Additionally, the stripes may make it harder for horseflies to land on zebras, as the stripes can disrupt the polarized light that horseflies use to find their prey.”

*Q2: How do zebras digest their food?*

Mistral says: “Zebras digest their food in the cecum, a blind sac at the far end of the small intestine. They are less efficient than ruminants in digesting grasses, but they compensate by ingesting more forage. The selective advantage of this approach is that zebras can subsist on range grasses unsuitable for antelope.”

*Q3: Tell me something about Grevy’s zebra.*

Mistral says: “Grevy's zebra (Equus grevyi) is one of the three species of strikingly black-and-white striped mammals of the horse family Equidae, found in arid, sparsely wooded areas in Kenya and a few small areas in Ethiopia. Adult males are larger than females, with spade-shaped canines used in fighting. The teeth of all three species of zebras are adapted for grazing, with strong upper and lower incisors for cropping grasses and large high-crowned teeth for processing silicate-rich grasses that wear down molars. Grevy's zebras do not form lasting bonds with each other, but related females may occupy the same grazing areas. Territoriality has evolved in Grevy's zebras because resources are widely scattered and easily defendable. With plentiful food, small groups may coalesce into large herds, but the smaller groups still retain their identities. Grevy's zebras often form mixed herds with other mammals such as wildebeests and giraffes, which gain protection from predators by the alertness of the zebras.”

Click here for broader view of images

```
Ask a question (or type 'exit'): Why Do Zebras Have Stripes? Explain in brief.

🧠 Mistral says: Zebras have stripes to help them blend in with their surroundings and avoid detection by pred
ators. The stripes make it difficult for predators to single out an individual zebra from a group. Additionall
y, the stripes may make it harder for horseflies to land on zebras, as the stripes can disrupt the polarized l
ight that horseflies use to find their prey.
```

Click here for broader view of images

```
Ask a question (or type 'exit'): How zebras digest their food?

🧠 Mistral says: Zebras digest their food in the cecum, a blind sac at the far end of the small intestine. The
y are less efficient than ruminants in digesting grasses, but they compensate by ingesting more forage. The se
lective advantage of this approach is that zebras can subsist on range grasses unsuitable for antelope.
```

Click here for broader view of images

```
Ask a question (or type 'exit'): Tell me something about Grevy's zebra.

🧠 Mistral says: Grevy's zebra (Equus grevyi) is one of the three species of strikingly black-and-white stripe
d mammals of the horse family Equidae, found in arid, sparsely wooded areas in Kenya and a few small areas in
Ethiopia. Adult males are larger than females, with spade-shaped canines used in fighting. The teeth of all th
ree species of zebras are adapted for grazing, with strong upper and lower incisors for cropping grasses and l
arge high-crowned teeth for processing silicate-rich grasses that wear down molars. Grevy's zebras do not form
 lasting bonds with each other, but related females may occupy the same grazing areas. Territoriality has evol
ved in Grevy's zebras because resources are widely scattered and easily defendable. With plentiful food, small
 groups may coalesce into large herds, but the smaller groups still retain their identities. Grevy's zebras of
ten form mixed herds with other mammals such as wildebeests and giraffes, which gain protection from predators
 by the alertness of the zebras.
```

For Code and Demo Visit - GitHub

*Observations:*

- The system provided factual, concise, and contextually relevant answers for all queries.
- The output quality was highly readable and informative, matching the level expected from modern LLMs.
- The use of a local Mistral-7B model proved effective even without internet-based APIs.
- Answers respected the scope of the scraped content, avoiding hallucination.

*Applications:*

- Academic Assistance: The tool can serve students by answering questions from academic articles, research papers, or encyclopedias.
- Enterprise Document Search: It can be integrated to process and answer questions over internal documentation or manuals.
- Offline Knowledge Assistant: The offline capability using a local LLM makes it suitable for environments with limited or no internet access.

## 5. Literature Review

Retrieval-Augmented Generation (RAG) has become a popular paradigm in modern NLP, where external knowledge sources are used to ground the responses of language models. Various studies and tools have explored this design:

1. Lewis et al. (2020) introduced the original RAG architecture that combines a retriever (like DPR or FAISS) with a generator (such as BART), enabling factual QA from external documents.
2. Haystack and LangChain provide modular RAG pipelines but typically require cloud APIs or GPUs, making them resource-intensive for individual or offline users.
3. Projects like OpenAI's GPT with plugins or Hugging Face's RAG pipelines allow similar contextual QA but rely on external models and APIs.
4. Local model loading via llama-cpp-python and gguf format has recently emerged as a viable alternative for inference without internet connectivity.

| Title / Tool | Description | Link |
|---|---|---|
| **RAG: Retrieval-Augmented Generation for Knowledge-Intensive NLP** *Lewis et al., 2020* | Foundation paper that introduced RAG, combining retriever | https://arxiv.org/abs/2005.11401 |
| **llama-cpp-python** | Lightweight backend to run LLaMA/Mistral models locally | https://github.com/abetlen/llama-cpp-python |
| **SentenceTransformers** | Python library for generating semantically meaningful sentence embeddings. | https://www.sbert.net/ |
| **FAISS (Facebook AI Similarity Search)** | Open-source vector store for fast similarity search and retrieval. | https://faiss.ai/ |

## 6. Peer Review

This project was successfully shaped with the guidance, feedback, and critical review provided by faculty members who played a key role throughout different stages of development. Each of them contributed uniquely to ensure the quality and credibility of this research paper.

*Prof. Makarand Patil*
Provided continuous mentorship, monitored the project timeline, and helped refine the research scope. His suggestions were pivotal in structuring the paper and ensuring academic alignment throughout the implementation phase.

*Prof. Anagha Chapadkar*
Played an instrumental role in reviewing the technical design, especially the architecture of the Retrieval-Augmented Generation pipeline. Her detailed feedback improved the flow of the System Design and Implementation sections.

*Prof. Swanand Pasalkar*
Offered critical insights into the evaluation and usability of the project. He also reviewed the Results section and recommended enhancements in the user interaction layer and response relevance testing.

Their combined efforts enriched this research by helping identify gaps, improve readability, and ensure that the technical foundations were sound. Their involvement served as valuable. peer-review process and enhanced the final outcome of the paper.

## 7. Conclusion and Future Work

In this project, we proposed and implemented a lightweight, context-aware retrieval-augmented generation (RAG) system powered by local LLMs. By integrating a dynamic web scraping pipeline, sentence embeddings with FAISS indexing, and inference through the Mistral model, our system is capable of answering user queries based on real-time contextual data from either URLs or text files.

The model has shown accurate and relevant responses, even without relying on cloud APIs, making it a cost-effective and secure solution for context-based question answering. The modular design—separating scraping, embedding, and inference—makes the system highly adaptable to different domains.

*For future improvements, we plan to:*
- Deploy a user-friendly web interface.
- Introduce summarization and multi-document support.
- Incorporate fine-tuned embeddings for better semantic matching.
- Explore integration with speech-to-text for voice queries.

## 8. References

[1] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP," arXiv preprint arXiv:2005.11401, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401
[2] deepset GmbH, "Haystack: Modular NLP Framework for Question Answering." [Online]. Available: https://haystack.deepset.ai/
[3] H. Chase et al., "LangChain Documentation." [Online]. Available: https://docs.langchain.com/
[4] abetlen, "llama-cpp-python: Python bindings for LLaMA inference." GitHub. [Online]. Available: https://github.com/abetlen/llama-cpp-python
[5] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," arXiv preprint arXiv:1908.10084, 2019. [Online]. Available: https://www.sbert.net/
[6] Facebook AI Research, "FAISS: A library for efficient similarity search and clustering of dense vectors." [Online]. Available: https://faiss.ai/
[7] TheBloke, "Mistral-7B-Instruct-v0.1 (GGUF version)," Hugging Face. [Online]. Available: https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.1-GGUF