Enhancing Retrieval Augmented Generation with Conversational Memory: A Domain-Grounded Context-Aware Architecture

Aneeqa Imtiyaz[1], Ms. Rupinder Kaur[2]

[1]*Student, Department of Computer Science and Engineering, Swami Vivekanand Institute of Engineering & Technology, Ramnagar, Banur, Punjab, India*

[2]*Assistant Professor, Department of Computer Science and Engineering, Swami Vivekanand Institute of Engineering & Technology, Ramnagar, Banur, Punjab, India*

***Abstract*—In the constantly evolving field of Natural Language Processing, Retrieval-Augmented Generation models have emerged as a powerful tool to enhance the contextual relevance of large language models. However, most RAG implementations lack long-term memory and session awareness, restricting their ability to hold meaningful multi-turn talks. This study proposes a database-driven, context-aware RAG architecture for improving conversational continuity by extracting top-N historical interactions from a MongoDB session database. These are integrated with document-based context obtained through vector search in ChromaDB. The final prompt, created with both document and session context, is sent to a Gemini 1.5 Pro model via API for generation. The suggested approach was tested across several sessions with six domain-specific papers. The results of evaluation showed high relevance of context and correct responses, especially in further requests. The proposed architecture can be seen as a bridge between traditional RAG systems and conversational agents that are capable of continuing sessions. It tries to introduce an efficient way for producing flexible responses.**

***Index Terms*—Context-awareness, Conversational AI, Large Language Model, Natural Language Processing, Retrieval-Augmented Generation, session memory.**

I. INTRODUCTION

Natural Language Processing (NLP) has experienced a tremendous development in the last several years, particularly due to Large Language Model (LLMs) that can produce rational and correct text within the context [1][2]. Nonetheless, the classical LLMs tend to lack continuity in multi-turn dialogue, especially in cases where the context spans over a large number of queries or sessions [3][4]. The current architectures of Retrieval-Augmented Generation (RAG) seek to overcome this through adding external knowledge which is specific with regard to a given domain into the generation process [5]. Although this enhances accuracy and reliability of information, majority of it cannot. RAG systems continue to operate without a conversational continuity, and there is no persistence in the past experiences of interactions [6][7].

This research paper fills the gap by proposing a context-aware RAG framework that is able to maintain session continuity. The proposed system stores user interactions in a *MongoDB* [8] database, retrieves the top-N most recent question-answer pairs during each prompt creation, and combines this session context with document-based context obtained using vector similarity. This compound prompt is passed to a *Gemini 1.5 Pro* LLM [9], which generates more logical, tailored, and context-rich responses.

The system was implemented using *LangChain* [10], *Hugging Face sentence transformers* [11], *ChromaDB* [12], and *Gradio* for interface deployment. To assess the system's flexibility, evaluations were undertaken in six different domains: *Machine learning, Psychology, History, Biology, General science,* and *Environmental science*. The results demonstrated enhanced contextual accuracy and response quality, especially in follow-up queries.

This study contributes to the growing field of adaptive language modeling by suggesting a practical architecture for incorporating long-term conversational memory into RAG systems.

II. LITERATURE SURVEY

The role of Neural Networks has been crucial in the progress of machine learning, especially after backpropagation techniques were introduced, that contributed in better optimization of weights in multi-layer percerptrons [4]. Early investigation on recurrent structures, like Elman networks [5], laid the foundation for handling sequential data, however, limitations such as, vanishing gradients continued to pose significant problems [6]. The introduction of Long Short-Term Memory (LSTM) networks addressed these problems with the help of gated memory cells, allowing the model to manage long-term dependencies more effectively [13]. The significance of LSTM and related models has been emphasized in sequence-to-sequence applications and their development into encoder-decoder architectures for effective understanding of natural language [7][14][15].

Transformer architecture was introduced to represent a major shift in the field of Natural Language Processing (NLP), through the integration of self-attention mechanisms to enhance scalability and representations of context in long-term sequences [1]. This architecture led to significant improvements across various NLP tasks by facilitating the introduction of large models like BERT and GPT [2][3][16]. However, a large number of models operate without preserving memory or maintaining state, so achieving meaningful dialogue across multiple conversations still remains a challenge.

Retrieval-Augmented Generation (RAG) frameworks have been developed to handle this issue, in which document retrieval supports the process of text generation. Traditional RAG systems [17] were successful in integrating external knowledge into response generation, but some studies have highlighted their limitations in maintaining coherence across entire sessions and ensuring clarity in evaluation methods [18][19][20]. Recent research has indicated that conventional evaluation metrics frequently fail to account for the influence of retrieved context, highlighting the need for new benchmarks to assess retrieval quality [21].

Context-Awareness has emerged as a vital factor in enhancing interactions across multiple turns. Research has suggested the incorporation of session memory into model prompts to improve relevance and coherence [22][23]. Context-Aware self-attention frameworks like CSAN [24] and strategies utilizing multi-source attention [25] present promising approaches for capturing dependencies in conversations. However, many systems still do not provide a cohesive strategy for merging document-grounded context with historical session data.

Review articles on context in NLP have further emphasized the challenges of scaling context-aware models across various domains [26][27]. Strategies such as Tri-attention [22] and models for dialogue that consider session-awareness [23] strive to incorporate longer dialogue histories, but often necessitate fine-tuning or domain-specific designs. This creates a research gap in the development of generic, domain-independent, session-persistent RAG systems.

The present work seeks to fill this gap by combining a vector-based document retriever (ChromaDB) with a session-level memory component (MongoDB). By integrating both the context of the document and prior Q&A pairs into one prompt, it is possible to receive more responsive and rational outputs from the LLM. The proposed system is constructed on the fundamental ideas of RAG, transfer learning [28][29], and recent researches on effective incorporation of context in LLMs [18][30].

III. PROPOSED SYSTEM

This section explains the composition and implementation of the context-aware Retrieval-Augmented Generation (RAG) system proposed that integrates session-based with document-based context to boost the answer generation in multi-turn communication.

A. System Overview

The primary aim of the suggested system is to provide adequate and proper answers, which incorporate not only the uploaded documents but also the history of the conversation. To do this, our system works with a combination of both the vector-based document retrieval and retrieval of the session memory stored in a database. Such a mixed context is used to generate a context-rich prompt that is subsequently fed into a large language model to generate responses as shown in Fig. 1.

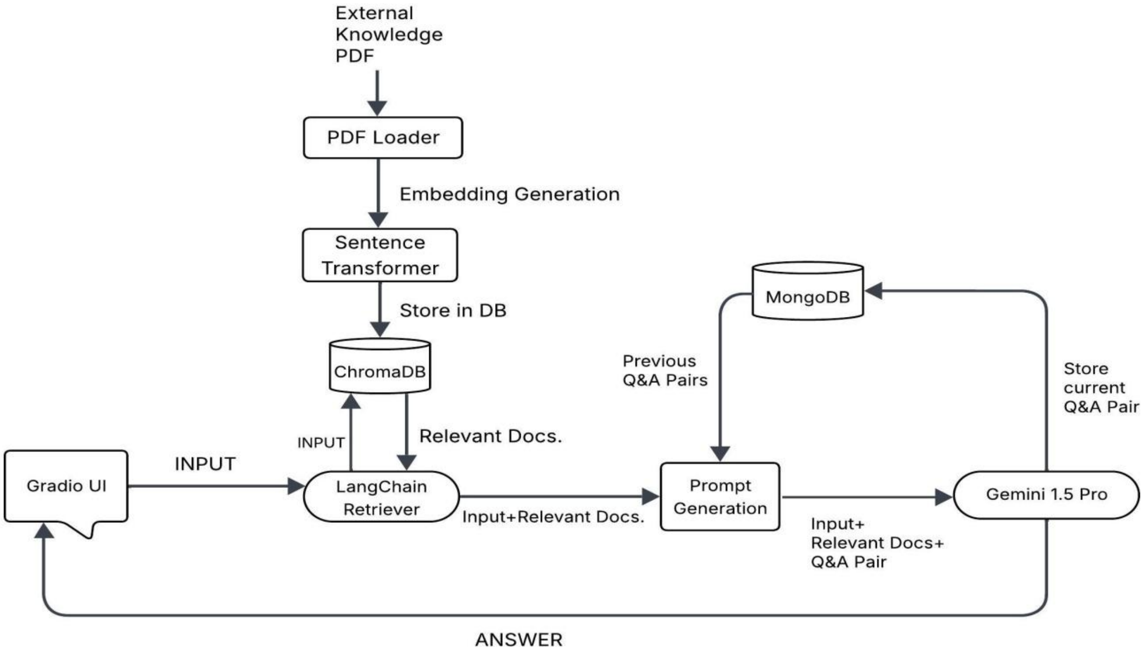

Fig. 1: Illustrates the system components and their interaction.

B. Environmental Setup And Dependencies

The proposed system was built and run using the following setup and dependencies:

1. Programming Language: *Python*
2. Platform Used: *Google Colab*
3. Libraries Used:

a) *LangChain* for chaining LLM-based tasks and managing retriever.
b) *HuggingFace Sentence-Transformers* for embedding generation.
c) *PyMongo* for *MongoDB*-based chat session storage.
d) *Gradio* for building the user interface.
e) *Uuid, requests*, and standard utility libraries.

C. System Workflow Description

The proposed system operates in the following stages:

1. Text preprocessing & chunking: Uploaded documents are analyzed and segmented into semantically meaningful text chunks. Sentence embeddings are created using a *Sentence-Transformer* model to maintain contextual similarity during the retrieval process
2. Document embedding and storage (ChromaDB): The embeddings that are generated are stored in *ChromaDB*, a vector database designed for efficient similarity searching. When a user poses a question, the system conducts a top-K similarity search to retrieve the most relevant document chunks.
3. Session context retrieval (MongoDB): To facilitate multi-turn context, recent question-answer pairs from the same session are stored in *MongoDB*. The system retrieves the top-N historical exchanges in reverse chronological order to mimic conversational memory.
4. Prompt construction: The retrieved document chunks and session history are combined into a single, structured prompt using a custom template. This prompt follows a format optimized for clarity and grounding, allowing the mode to differentiate between background information and the current inquiry.
5. Response generation (Gemini 1.5 pro): The consolidated prompt is sent to the *Gemini 1.5 Pro* language model via the *Google Cloud API*. The model generates a response that is expected to be accurate, contextually relevant, and grounded in the content of the uploaded documents.

6. Response display and logging: The generated response is shown to the user through *Gradio* interface. At the same time, both the question and the answer are recorded in the *MongoDB* session log for use in future context.

D. *Prompt Flow Design*

The system departs from conventional *LangChain* templates and instead employs a custom prompt structure. This format includes:

1. System role definition (guidance for the model)
2. Session history (prior n QA pairs)
3. Retrieved chunks from the vector database
4. Present user question

This layered approach guarantees that the LLM can consider both static document knowledge and dynamic user interaction context when formulating its response.

E. *Workflow Summary*

The entire workflow is summarized as follows:

1. A domain-specific document PDF is uploaded.
2. The document is chunked, embedded, and stored.
3. The user submits a question.
4. Relevant document chunks and session memory are retrieved.
5. A combined prompt is generated.
6. A response is created and displayed to the user
7. The QA pair is recorded for subsequent queries

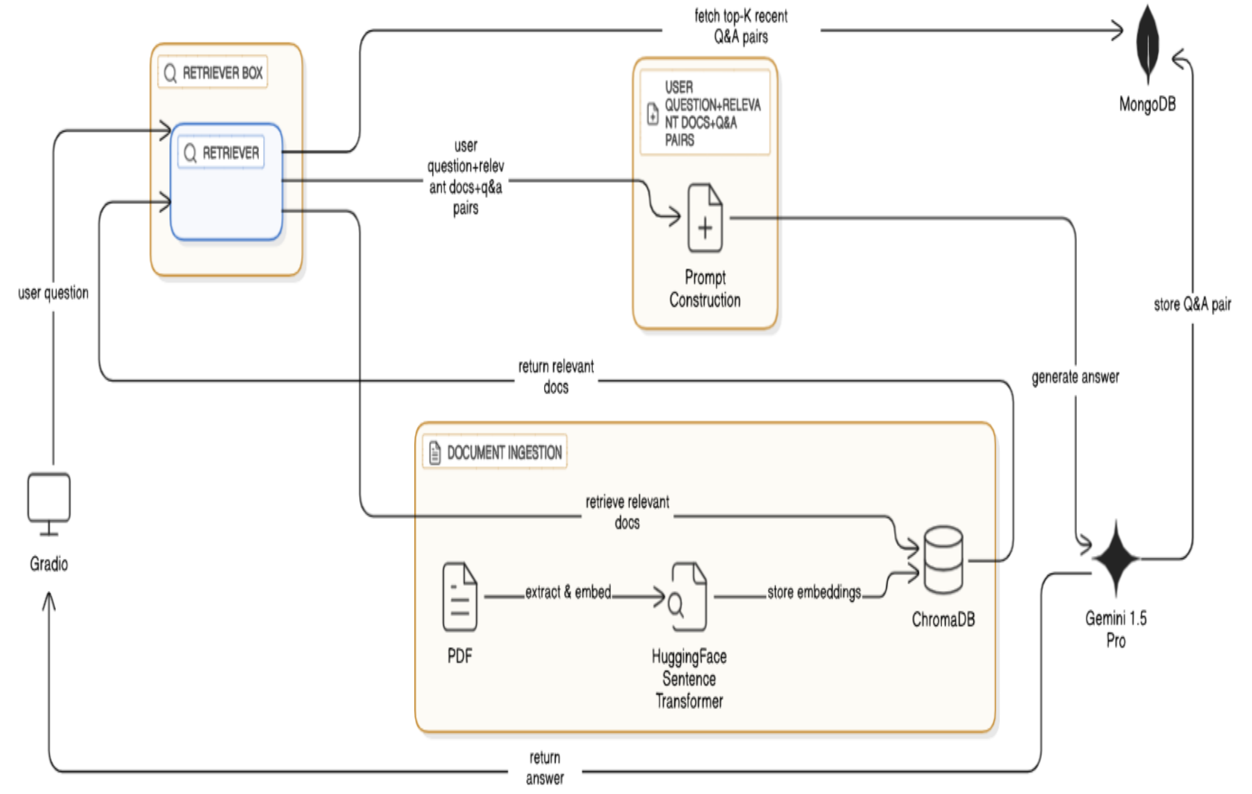

Fig. 2: Workflow of proposed Context-aware RAG system.

This system facilitates context-aware QA without the necessity of retraining the model from scratch or altering its architecture. It can easily scale across multiple domains by uploading new documents, making it highly adaptable for academic, technical, and general knowledge applications. The workflow description of the proposed model is shown in Fig. 2.

IV. EVLUATION DESIGN AND STRATEGY

To assess the effectiveness and adaptability of the suggested context-aware RAG system, a structured multi-session testing strategy was implemented. This included utilizing domain-specific PDFs, manually crafted queries, and comparative assessments between a standard RAG configuration and the improved context-aware version.

A. Domain Documents

A set of six unique domain-specific documents was employed to evaluate the system's capacity to ground responses in relevant knowledge and dynamically adapt across various domains as shown in Table 1. These documents encompassed both academic and non-academic PDFs, featuring diverse styles and complexities. These documents were uploaded sequentially during the testing phases to replicate real-world situations where the system is expected to function with a single knowledge source at any given time.

Table I: Table showing PDFs and their Domain Characteristics.

DOMAINS	DOCUMENT SOURCE	TYPE
MACHINE LEARNING	B. Tech IV Year ML PDF	TECHNICAL
PSYCHOLOGY	"Introduction to Psychology" (2021 ed.)	THEORETICAL/ NARRATIVE
BIOLOGY	NCERT Class 12 Biology	ACADEMIC TEXTBOOK
GENERAL SCIENCE	Maharashtra State General Science	NARRATIVE/ DESCRIPTIVE
HISTORY	AFEIAS History Notes (English)	BASIC SCIENCE TEXT
ENVIRONMENTAL SCIENCE	Vardhaman College Env. Studies	CURRICULUM-BASED TEXT

B. Testing Sessions and Query Design

Each domain was assessed through a separate session, enabling the context-aware memory mechanism. The sessions were structured to include different types of question as shown in Fig. 3. These include:

1. Clearly addressable questions (answers directly available in the document).
2. Slightly relevant questions (content that overlaps indirectly or partially).
3. Unrelated/out-of-scope questions (e.g., general inquiries or cross-topic questions).

For each session:

1. 5-10 queries were posted in order.
2. The model's replies were monitored for coherence, factual accuracy, and memory consistency.
3. Session records were kept and reviewed.

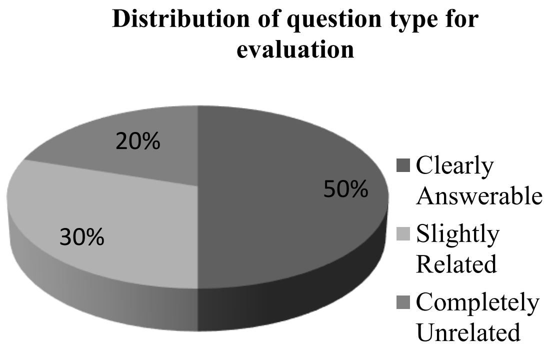

Fig. 3: The pie chart illustrates the question category distribution used to assess system behavior under varied conditions.

C. Baseline RAG vs. Context-Aware RAG

To evaluate the impact of session memory integration, two configurations were compared:

1. Baseline RAG: Only document chunks were retrieved using ChromaDB, with no memory of previous QA pairs.
2. Context-Aware RAG: Merged document retrieval with top-N past question-answer pairs retrieved from MongoDB session records.

This arrangement enabled a direct assessment of response quality in both single-turn and multi-turn scenarios.

D. Evaluation Criteria

All responses were manually evaluated using three primary metrics:

1. Contextual Relevance Score (CRS): Rated on a scale of 1-5, reflecting how well the answer was supported by document and session context.
2. Answer Accuracy (AA): The percentage of responses that were factually correct according to the references document.

3. Response Coherence (RC): Rated from 1-5, evaluating fluency, consistency, and logical flow between turns.

Each answer was graded by two evaluators following a fixed rubric to ensure uniformity. The evaluation considered both adaptability to the domain and retention of context.

V. RESULTS AND DISCUSSION

This section outlines the evaluation findings of the introduced context-aware Retrieval-Augmented Generation (RAG) system, contrasting it with a standard RAG implementation. The comparison centers on the impact of session memory on response accuracy, contextual relevance, and coherence during multi-turn question-answering.

A. Performance Metrics

The three-evaluation metrics applied to both system variants were:

1. Contextual Relevance Score (CRS).
2. Answer Accuracy (AA).
3. Response Coherence (RC).

The average outcomes across six domain-specific sessions are displayed in table II.

Table II: Quantitative comparison of Baseline and Session-Aware RAG systems using custom evaluation metrics.

Metric Used	Baseline RAG	Context-Aware RAG
Contextual Relevance Score	2.6/5	4.5/5
Answer Accuracy	62%	85%
Response Coherence	2.6/5	4.5/5

The scores indicate a substantial improvement in performance when session memory is utilized.

The graph comparing the scores of baseline RAG and Context-Aware RAG is shown in Fig.4.

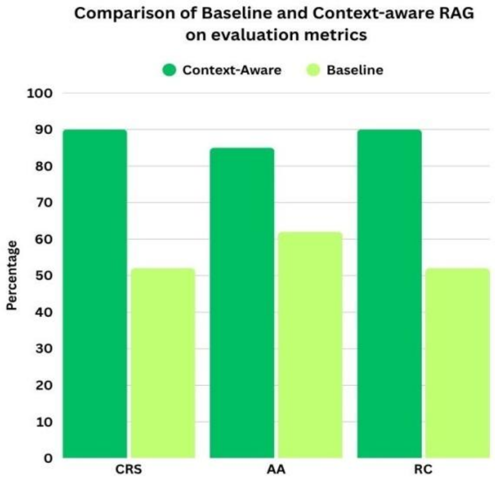

Fig. 4: Bar graph comparing CRS, AA, and RC - Baseline vs. Context-Aware

B. Observations and Insights

The following observations have been noted in the process of analysis of the proposed model:

1. Enhanced follow-up management: The context-aware system consistently produced superior answer to follow-up queries. The inclusion of previous Q&A pairs helped sustain logical continuity across turns.
2. Decreased hallucination risk: By grounding responses solely in the content retrieved from documents and prior queries, the model successfully avoided generating hallucinated or irrelevant outputs, particularly for unrelated queries.
3. Adaptability across domains: The proposed system was also flexible in terms of adapting to various topics since the only thing that needed to be changed was the uploaded PDF and not any retraining of the model. This brings to the fore the modularity of RAG framework.
4. Fall-back responses: The model well reacted with fall back answers such as "*I don't know*" to questions that fall outside its scope, refraining from giving irrelevant or false information.
5. Consistency in evaluation: Ratings were stable among evaluators, indicating consistent system behavior across all six testing domains.

C. Limitations

Although the proposed system yielded encouraging results, it does have several limitations:

1. Manual evaluation: The assessment was performed manually in the absence of automated metrics such as BLEU or ROUGE, which could introduce a degree of subjectivity.
2. Session limitation: Only the top-N historical Q&A pairs were considered managing longer session histories may necessitate enhanced memory management.
3. Dependence on language model: The quality of the generated responses is partially reliant on the capabilities of the Gemini LLM.

VI. CONCLUSION AND FUTURE WORK

The paper presented a context-aware Retrieval-Augmented Generation (RAG) model that is beneficial in enhancing long-range coherence and factual accuracy in a multi-turn question-answering scenario. The model uses retrieved documents available in ChromaDB in combination with session-level context data available in MongoDB, to create prompts that have context, yet continue across conversation. Gemini 1.5 Pro LLM usage allows the system to generate high-quality responses, and modularity provided by such tools as LangChain, Hugging Face Sentence Transformers, and Gradio allow scalability and domain flexibility.

The assessment carried out in six domain-specific document session indicated the reliability of the system in terms of showing considerable accuracy, contextual relevance, and coherence compared to a conversational RAG framework. The manual testing technique and assessment criteria-CRS, Answer Accuracy, and Response Coherence-were able to give important information regarding the efficiency off session-aware memory in real-time applications.

The proposed architecture effectively integrates traditional RAG systems and conversational agents; however, it does have some limitations. Since the evaluation process was time-consuming, the use of automated metrics like ROUGE or BERTSCORE could offer a more scalable evaluation strategy. Moreover, the system currently focuses on textual input: expanding it to include multimodal retrieval or cross-lingual features offers a promising direction for future enhancement.

The aim of future research will be focused on enhancing dynamic memory for long sessions, adjusting vector retrieval threshold to improve performance, and exploring the implementation of the proposed system in real-time assistant applications. Applying reinforcement learning to refine prompts adaptively and updating memory with user inputs can further help in maintaining coherence in conversations and adapting to individual users

REFERENCES

[1] Vaswani *et al.*, “Attention Is All You Need,” *Advances in Neural Information Processing Systems*, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[2] T. Brown *et al.*, “Language Models Are Few-Shot Learners,” *Advances in Neural Information Processing Systems*, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Representations by Back-Propagating Errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986. [Online]. Available: https://doi.org/10.1038/323533a0

[5] J. L. Elman, “Finding Structure in Time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990. [Online]. Available: https://doi.org/10.1207/s15516709cog1402_1

[6] R. Pascanu, T. Mikolov, and Y. Bengio, “On the Difficulty of Training Recurrent Neural Networks,” in *Proc. Int. Conf. Machine Learning (ICML)*, 2013. [Online]. Available: https://arxiv.org/abs/1211.5063

[7] G. Van Houdt, C. Mosquera, and G. Nápoles, “A Review on the Long Short-Term Memory Model,” *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020. [Online]. Available: https://doi.org/10.1007/s10462-020-09838-1

[8] MongoDB Inc., “MongoDB – Document-Oriented NoSQL Database System.” [Online]. Available: https://www.mongodb.com

[9] Google Cloud, “Gemini 1.5 Pro via ChatGoogleGenerativeAI API.” [Online]. Available: https://cloud.google.com/vertex-ai/generative-ai/docs

[10] LangChain, “LangChain – Framework for Developing Applications Powered by Language Models.” [Online]. Available: https://www.langchain.com

[11] Hugging Face, “Sentence Transformers – Pretrained Models for Sentence Embeddings.” [Online]. Available: https://www.sbert.net

[12] Chroma, “ChromaDB – Open-Source Vector Database for Embedding Retrieval.” [Online]. Available: https://www.trychroma.com

[13] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[14] Y. Zhang, “Encoder Decoder Models in Sequence-to-Sequence Learning: A Survey of RNN and LSTM Approaches,” *Applied and Computational Engineering*, vol. 22, no. 1, pp. 218–226, 2023. [Online]. Available: https://doi.org/10.54254/2755-2721/22/20231220

[15] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” in *Advances in Neural Information Processing Systems*, 2014. [Online]. Available: https://arxiv.org/abs/1409.3215

[16] W. X. Zhao *et al.*, “A Survey of Large Language Models,” *arXiv preprint*, arXiv:2303.18223, 2023. [Online]. Available: https://arxiv.org/abs/2303.18223

[17] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in *Advances in Neural Information Processing Systems*, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401

[18] Y. Gao *et al.*, “Retrieval-Augmented Generation for Large Language Models: A Survey,” *arXiv preprint*, arXiv:2405.07437, 2024. [Online]. Available: https://arxiv.org/abs/2405.07437

[19] H. Yu *et al.*, “Evaluation of Retrieval-Augmented Generation: A Survey,” *arXiv preprint*, arXiv:2405.07437, 2024. [Online]. Available: https://arxiv.org/abs/2405.07437

[20] H. Li *et al.*, “A Survey on Retrieval-Augmented Text Generation,” *arXiv preprint*, arXiv:2202.01110, 2022. [Online]. Available: https://arxiv.org/abs/2202.01110

[21] A. Salemi and H. Zamani, “Evaluating Retrieval Quality in Retrieval-Augmented Generation,” in *Proc. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2024. [Online]. Available: https://doi.org/10.1145/3626772.3657957

[22] R. Yu *et al.*, “Tri-Attention: Explicit Context-Aware Attention Mechanism for NLP,” 2022. [Online]. Available: https://github.com/yurui12138/Tri-Attention

[23] A. Gupta *et al.*, “CASA-NLU: Context-Aware Self-Attentive Natural Language Understanding for Task-Oriented Chatbots,” *arXiv preprint*, arXiv:1909.08705, 2019. [Online]. Available: https://arxiv.org/abs/1909.08705

[24] B. Yang *et al.*, “Context-Aware Self-Attention Networks,” in *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, vol. 33, no. 01, pp. 387–394, 2019. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.3301387

[25] H. H. Vu, H. Kamigaito, and T. Watanabe, “Context-Aware Machine Translation with Source Coreference Explanation,” *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 856–874, 2024. [Online]. Available: https://doi.org/10.1162/tacl_a_00677

[26] N. Matta, N. Matta, and P. Herr, “Importance of Context Awareness in NLP,” in *Proc. Int. Conf. Knowledge Management and Information Systems (KDIR)*, 2024. [Online]. Available: https://doi.org/10.5220/0012994700003838

[27] A. Ignise and Y. Vahi, “Context Awareness Challenges in Natural Language Processing,” 2024.

[28] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A Survey of Transfer Learning,” *Journal of Big Data*, vol. 3, no. 1, May 2016. [Online]. Available: https://doi.org/10.1186/s40537-016-0043-6

[29] L. Torrey and J. Shavlik, “Transfer Learning,” in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, IGI Global, 2009, pp. 242–264. [Online]. Available: https://www.irma-international.org/chapter/transfer-learning/36988

[30] P. Zhao *et al.*, “Retrieval-Augmented Generation for AI-Generated Content: A Survey,” *arXiv preprint*, arXiv:2402.19473, 2024. [Online]. Available: https://arxiv.org/abs/2402.19473