VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Programming Intergration Project (CO3101)

Group 4:
"Research and build AI chatbots using Retrieval-Augmented Generation for a music-related website, with Speech-to-Text and Text-to-Speech integration"

Instructor(s): Nguyễn Quốc Minh

Students: Nguyễn Thiện Minh - 2312097
Huỳnh Đức Nhân - 2312420
Phạm Trần Minh Trí - 2313622

HO CHI MINH CITY, DECEMBER 2025

Contents

List of Figures

List of Tables

Member list & Workload

No.	Fullname	Student ID	Problems	% done
1	Nguyễn Thiện Minh	2312097	- Exercise 1: 1.2 - Exercise 2 - Exercise 3: 3.2	100%
2	Huỳnh Đức Nhân	2312420	- Exercise 1: 1.3 - Exercise 2 - Exercise 3: 3.1 - Exercise 4	100%
3	Phạm Trần Minh Trí	2313622	- Exercise 1: 1.1 - Exercise 4 - LaTeX	100%

Table 1: Member list & workload

1 Introduction

2 Prerequisite: ANN, Transformer, LLM

2.1 Artificial neural network

2.2 Transformer architecture

2.3 Large language model

Large Language Models (LLMs) are advanced neural network models designed to understand, generate, and manipulate natural language at scale. They are typically built upon the Transformer architecture and trained on massive text corpora, enabling them to perform a wide range of Natural Language Processing (NLP) tasks such as text generation, question answering, summarization, and dialogue systems.

2.3.1 Evolution and Architecture

The development of LLMs has progressed from traditional statistical language models (e.g., n-gram models) to recurrent architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, and ultimately to Transformer-based models. Transformers address the limitations of sequential computation and long-term dependency learning through the self-attention mechanism, making them highly scalable and effective for large datasets.

Based on their architectural design, LLMs can be categorized into three main types:

- **Decoder-only models (causal language models):** These models, such as GPT and LLaMA, predict the next token in a sequence and are primarily used for text generation tasks.
- **Encoder-only models (bidirectional language models):** Models like BERT, RoBERTa, and DistilBERT focus on learning contextual representations of text and are widely used for text understanding tasks such as classification and semantic similarity.
- **Encoder–Decoder models (sequence-to-sequence):** Examples include T5 and BART, which encode the input sequence and then decode it into an output sequence, making them suitable for machine translation, summarization, and question answering.

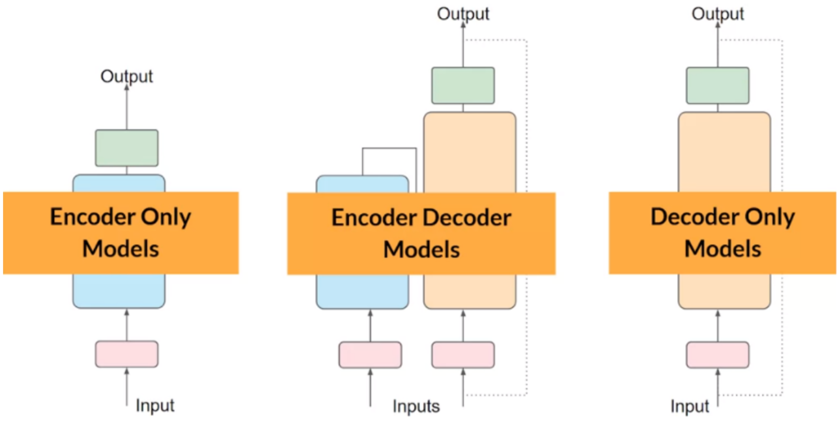


Figure 1: Transformer architectures

2.3.2 Pretraining and Fine-Tuning

LLMs are first trained during a pretraining phase using large-scale datasets such as Common Crawl, Wikipedia, and book corpora. Common pretraining objectives include autoregressive language modeling, masked language modeling, and denoising tasks. These objectives allow the model to learn grammar, semantics, and world knowledge from raw text.

After pretraining, LLMs are adapted to specific tasks through fine-tuning. This process may include Supervised Fine-Tuning (SFT), where human-labeled prompt–response pairs are used, and Reinforcement Learning from Human Feedback (RLHF), where human preferences guide the optimization of model outputs. Parameter-Efficient Fine-Tuning (PEFT) techniques such as LoRA and QLoRA are often employed to reduce computational cost while maintaining performance.

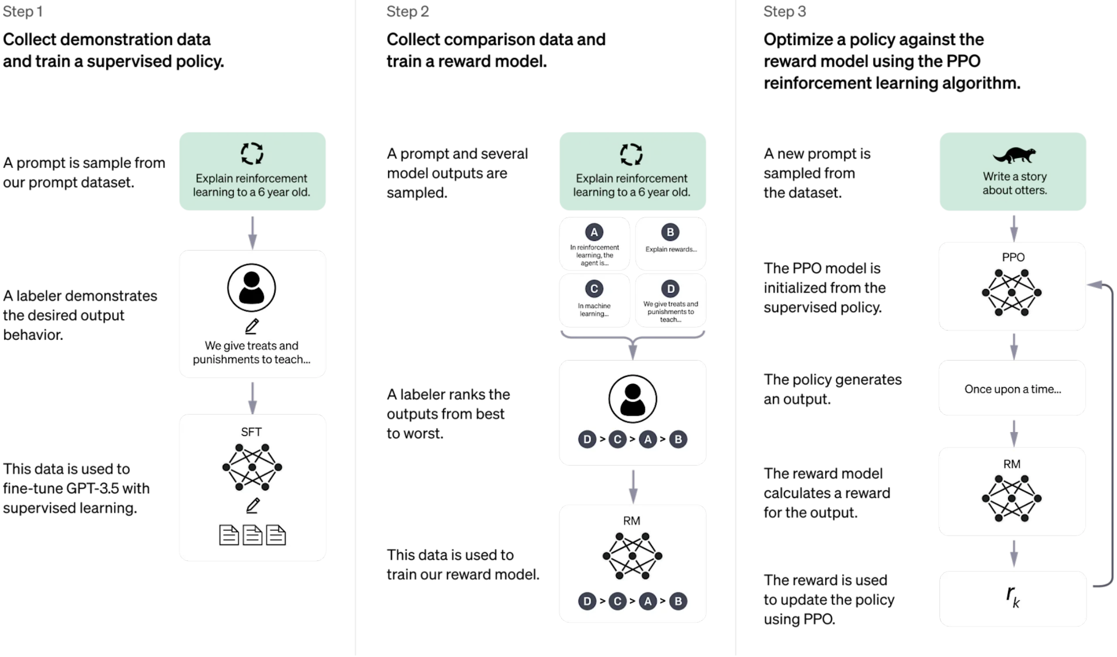

Step 1
Collect demonstration data and train a supervised policy.
A prompt is sample from our prompt dataset.
Explain reinforcement learning to a 6 year old.
A labeler demonstrates the desired output behavior.
We give treats and punishments to teach...
This data is used to fine-tune GPT-3.5 with supervised learning.
SFT
Step 2
Collect comparison data and train a reward model.
A prompt and several model outputs are sampled.
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
A labeler ranks the outputs from best to worst.
D > C > A > B
This data is used to train our reward model.
RM
D > C > A > B
Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.
A new prompt is sampled from the dataset.
Write a story about otters.
The PPO model is initialized from the supervised policy.
PPO
The policy generates an output.
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

Figure 2: RLHF for ChatGPT

2.3.3 LLMs in Chatbot Systems

LLMs play a central role in modern chatbot systems. To improve factual accuracy and domain specificity, Retrieval-Augmented Generation (RAG) is commonly used. In a RAG-based system, relevant documents are embedded into a vector space and stored in a vector database. When a user submits a query, the system retrieves the most relevant documents and provides them as additional context to the LLM before generating a response. This approach significantly enhances the reliability and explainability of chatbot answers.

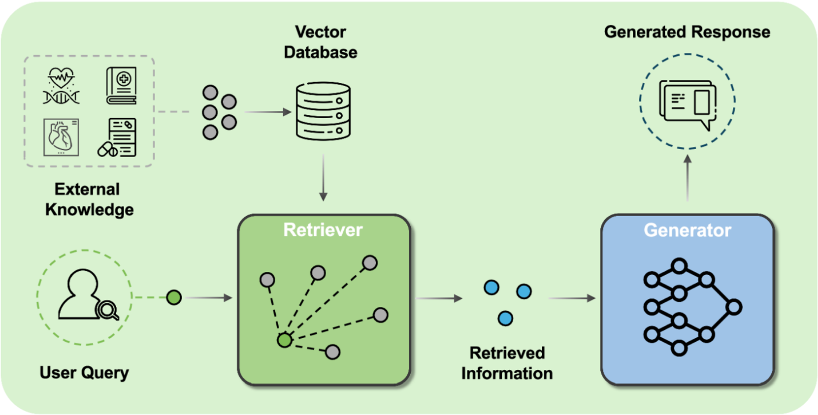


Figure 3: Retrieval augmented generation

In addition, prompt engineering and system instructions are used to control the behavior, style, and scope of the chatbot. For domain-specific applications such as a music information website, these techniques ensure that responses remain relevant, concise, and aligned with predefined constraints.

3 Agent, URAG, LangChain

3.1 AI agent

3.2 Unified Hybrid RAG

3.3 Introduction to LangChain

LangChain is a comprehensive framework designed to support the development, deployment, and monitoring of applications powered by Large Language Models (LLMs). It provides modular components and abstractions that simplify the construction of complex LLM-based systems such as Retrieval-Augmented Generation (RAG) pipelines and agentic workflows.

LangChain supports the full lifecycle of an LLM application. During the development phase, developers can build applications using LangChain's core components, including prompt templates, chains, retrievers, vector stores, and integrations with third-party tools and APIs. For more advanced agentic behaviors, LangChain introduces LangGraph, which enables the definition of multi-step agents with explicit states, nodes, and control flows. This graph-based design allows agents to reason, invoke tools, and iterate over multiple steps in a structured and controllable manner.

In the production phase, LangChain is complemented by LangSmith, a platform that provides observability, debugging, and evaluation capabilities. LangSmith allows developers to inspect prompt execution, track intermediate steps, monitor latency and costs, and systematically evaluate model outputs. These features are particularly important for RAG systems, where both retrieval quality and generation accuracy must be continuously assessed.

For deployment, LangChain offers the LangGraph Platform, which facilitates the deployment and scaling of agentic workflows. This platform-oriented approach makes LangChain suitable not only for experimentation but also for real-world applications.

Within a RAG architecture, LangChain structures the workflow into two main stages. The first stage is indexing, an offline process that involves loading documents, splitting them into chunks, generating embeddings using an embedding model, and storing these embeddings in a vector database. The second stage is retrieval and generation, which occurs at runtime: relevant document chunks are retrieved from the vector store based on the user query, combined with the query into a structured prompt, and passed to an LLM to generate a final response.

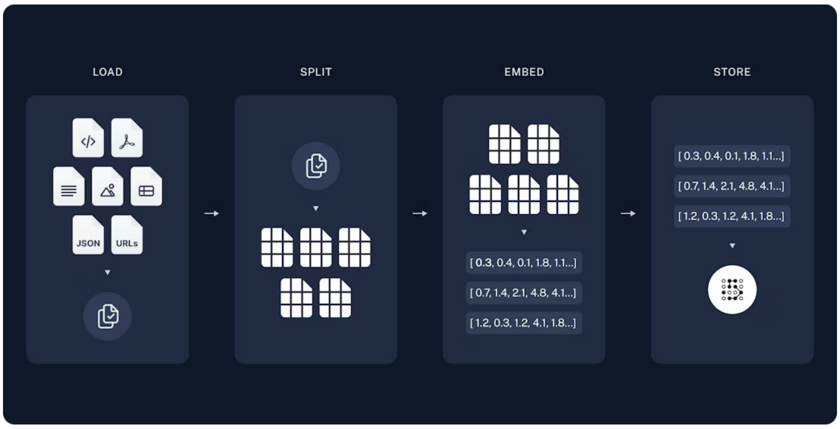


Figure 4: RAG - Indexing

LangChain provides a flexible and extensible foundation for building RAG-based and agent-driven applications, enabling developers to integrate LLMs, retrieval systems, and external tools into a unified and production-ready framework.

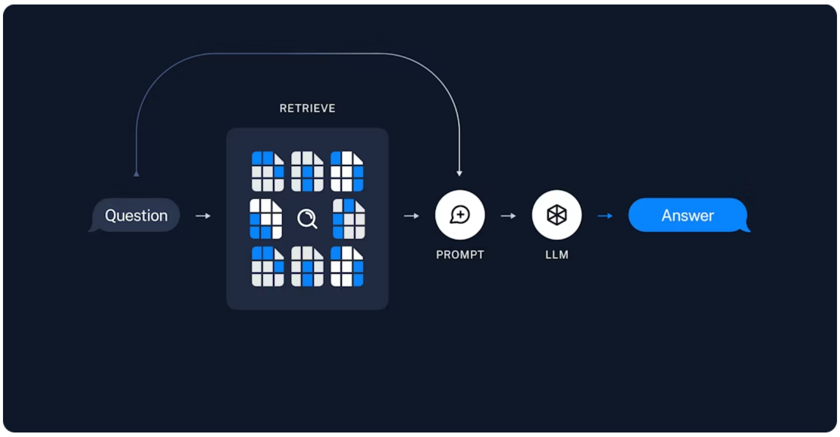


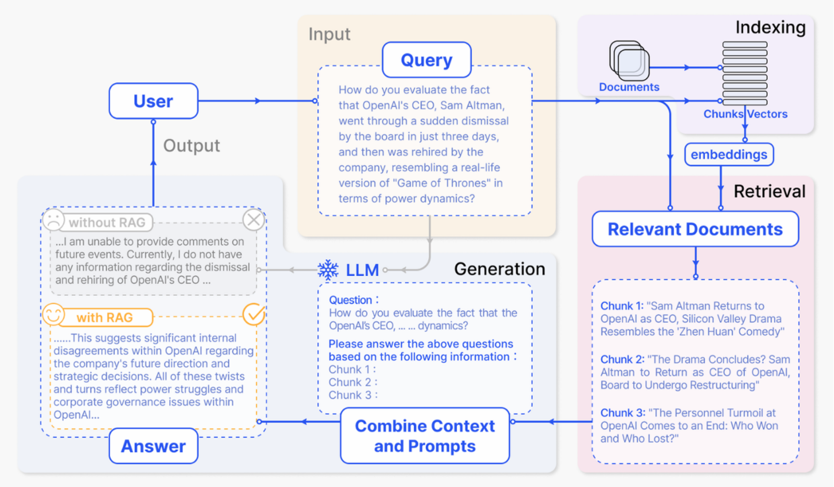
Figure 5: RAG - Retrieval and Generation

4 Retrieval-Augmented Generation for Large Language Models: A Survey

Link to the paper: https://arxiv.org/abs/2312.10997 [1]

4.1 Overview of RAG

Retrieval-Augmented Generation (RAG) is a paradigm that enhances large language models (LLMs) by incorporating external knowledge retrieved at inference time. Instead of relying solely on parametric knowledge stored in model weights, RAG dynamically retrieves relevant documents and uses them as additional context for response generation.



How do you evaluate the fact that OpenAI's CEO, Sam Altman, went through a sudden dismissal by the board in just three days, and then was rehired by the company, resembling a real-life version of "Game of Thrones" in terms of power dynamics?

...I am unable to provide comments on future events. Currently, I do not have any information regarding the dismissal and rehiring of OpenAI's CEO ...

......This suggests significant internal disagreements within OpenAI regarding the company's future direction and strategic decisions. All of these twists and turns reflect power struggles and corporate governance issues within OpenAI...

Question :
How do you evaluate the fact that the OpenAI's CEO, ... dynamics?
Please answer the above questions based on the following information :
Chunk 1 :
Chunk 2 :
Chunk 3 :

Chunk 1: "Sam Altman Returns to OpenAI as CEO, Silicon Valley Drama Resembles the 'Zhen Huan' Comedy"

Chunk 2: "The Drama Concludes? Sam Altman to Return as CEO of OpenAI, Board to Undergo Restructuring"

Chunk 3: "The Personnel Turmoil at OpenAI Comes to an End: Who Won and Who Lost?"

Figure 6: RAG overview

4.1.1 Naive RAG

The Naive RAG framework follows a simple *Retrieve–Read* pipeline consisting of three main stages: indexing, retrieval, and generation. During indexing, raw data sources such as PDFs, HTML pages, or Word documents are converted into plain text, segmented into smaller chunks, and encoded into vector representations stored in a vector database. In the retrieval stage, the user query is embedded and compared with stored vectors using similarity metrics to obtain the top-K most relevant chunks. Finally, in the generation stage, the LLM produces an answer based on the user query and the retrieved context, optionally incorporating conversation history.

Despite its simplicity, Naive RAG suffers from several limitations. Retrieval may lack precision and recall, resulting in irrelevant or missing information. During generation, the model may hallucinate content not supported by the retrieved context or produce outputs with bias or irrelevance. Moreover, effectively integrating retrieved information across different tasks remains challenging, often leading to redundant, incoherent, or overly extractive responses.

4.1.2 Advanced RAG

Advanced RAG aims to improve retrieval quality and context utilization through enhanced pre-retrieval and post-retrieval techniques. Pre-retrieval optimization focuses on improving indexing structures and query formulation, including data granularity control, metadata alignment, mixed retrieval strategies, and query rewriting or expansion. Post-retrieval optimization emphasizes effective context integration, such as re-ranking retrieved documents to prioritize relevance and compressing context to reduce noise and prompt length.

4.1.3 Modular RAG

Modular RAG extends beyond fixed retrieval-generation pipelines by introducing specialized, interchangeable modules. These include search modules for heterogeneous data sources, RAG-Fusion for multi-query expansion and re-ranking, and memory modules that maintain a continuously updated retrieval memory pool. Additional components such as routing, prediction, and task adapters allow RAG systems to dynamically select retrieval pathways and adapt to downstream tasks.

This modular design enables flexible retrieval patterns, including Rewrite–Retrieve–Read, Generate–Read, and hybrid retrieval strategies that combine keyword-based, semantic, and vector-based search. As a result, Modular RAG exhibits strong adaptability and scalability across diverse applications.

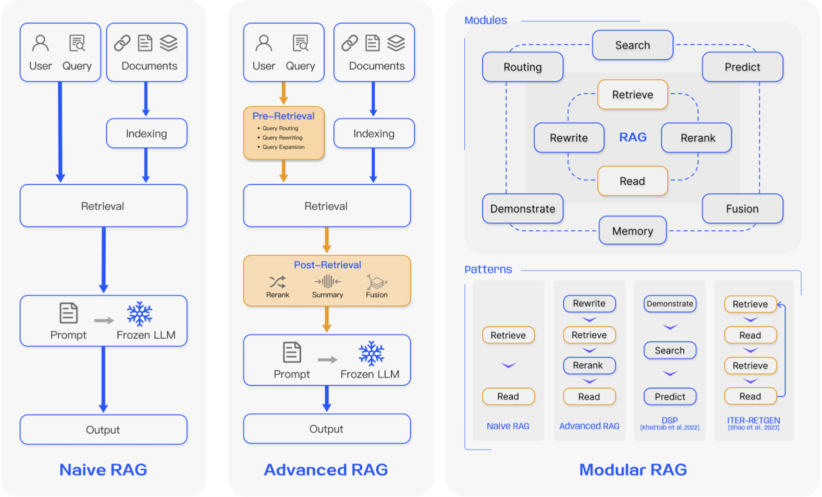


Figure 7: Types of RAG

4.2 Retrieval

Retrieval is a core component of RAG, responsible for identifying and supplying relevant external knowledge to the generation model. The effectiveness of a RAG system heavily depends on retrieval source selection, indexing strategies, query optimization, and embedding quality.

4.2.1 Retrieval Sources and Granularity

Retrieval sources can be categorized into unstructured, semi-structured, and structured data. Unstructured text data, such as Wikipedia articles or domain-specific documents, is the most common source. Semi-structured data, including PDFs with tables, presents challenges due to structural complexity, while structured sources like knowledge graphs offer precise and verified information at the cost of higher construction and maintenance effort.

Retrieval granularity ranges from tokens and sentences to chunks and full documents. Coarse-grained retrieval provides richer context but may introduce redundancy and noise, whereas fine-grained retrieval improves precision but risks losing essential semantic information.

4.2.2 Indexing Optimization

Indexing optimization techniques aim to balance context richness and efficiency. Chunking strategies play a critical role, where large chunks capture broader context but increase noise and computational cost, while small chunks reduce noise but may lack sufficient information. The *Small-to-Big* approach mitigates this trade-off by retrieving smaller units and expanding context hierarchically.

Metadata attachments, such as page numbers or timestamps, enable filtered retrieval and scoped search. Structural indexing methods, including hierarchical document structures and knowledge graph indices, further enhance retrieval speed and relevance. Techniques like Reverse HyDE leverage LLMs to generate potential questions that each chunk can answer, improving retrievability.

4.2.3 Query Optimization

Query optimization improves retrieval effectiveness by refining or expanding user queries. Query expansion and multi-query techniques enrich the query with additional context, while sub-query decomposition breaks complex questions into simpler ones. Query transformation methods include rewriting queries, generating hypothetical answers (HyDE), and step-back prompting to retrieve higher-level contextual information.

Query routing mechanisms further enhance retrieval by directing queries to appropriate data sources or pipelines using metadata-based or semantic routing strategies.

4.2.4 Embeddings and Adapters

Modern RAG systems often employ hybrid retrieval that combines sparse retrievers, such as BM25 for keyword matching, with dense retrievers based on neural embeddings for semantic understanding. Embedding models can be fine-tuned for domain-specific tasks, with LM-supervised retrievers aligning retrieval objectives with generation outcomes using LLM feedback.

When fine-tuning is impractical, adapter-based methods provide lightweight alternatives. These include prompt retrievers, bridging modules that transform retrieved content into LLM-friendly formats, and plug-in knowledge generators that replace or augment traditional retrievers in white-box settings.

4.3 Generation

4.4 Augmentation

4.5 Task & evaluation

4.6 Discussion

5 Reranking, RAG-Reasoning, RAG-RL

5.1 Reranking in RAG

5.2 Towards Agentic RAG with Deep Reasoning: A Survey of RAG Reasoning Systems in LLMs

Link to paper: https://arxiv.org/abs/2507.09477 [5]

Retrieval-Augmented Generation (RAG) has emerged as a powerful paradigm for enhancing large language models (LLMs) by grounding generation in external knowledge sources. However, traditional RAG pipelines primarily focus on surface-level semantic retrieval and often struggle with multi-hop reasoning, noisy contexts, and complex decision-making tasks. The survey by Li et al. proposes a comprehensive framework that systematically analyzes how reasoning capabilities can be deeply integrated into RAG systems, moving toward more agentic and intelligent architectures.

The survey categorizes existing approaches into three major paradigms. The first is Reasoning-Enhanced RAG, where reasoning is explicitly incorporated to improve retrieval, integration, and generation. At the retrieval stage, techniques such as reasoning-aware query reformulation, retrieval planning, and retriever model enhancement aim to obtain evidence that is more relevant to downstream reasoning tasks. During integration, retrieved documents are assessed, filtered, and fused using reasoning-driven relevance assessment and information synthesis mechanisms. At the generation stage, context-aware and grounded generation methods ensure that the model's outputs remain faithful to retrieved evidence and follow coherent reasoning paths.

The second paradigm is RAG-Enhanced Reasoning, which treats retrieval as a tool to directly support the reasoning process of LLMs. In this setting, models retrieve external knowledge such as structured knowledge bases, web content, tools, or prior experiences to assist in complex reasoning tasks including mathematical problem solving, fact verification, and planning. In-context retrieval of examples and historical interactions further enables LLMs to adapt reasoning strategies dynamically based on retrieved demonstrations or memories.

The third paradigm, Synergized RAG-Reasoning, represents the most advanced integration, where retrieval and reasoning are tightly interwoven within an agentic workflow. These systems interleave reasoning steps with retrieval actions using chain-based, tree-based, or graph-based reasoning workflows. Moreover, agent orchestration techniques, including single-agent and multi-agent systems, allow LLMs to autonomously plan, retrieve, reason, and verify information. Such agentic RAG systems exhibit improved robustness, interpretability, and adaptability in complex tasks.

Category	**Method summary**	**Related papers**
Reasoning-Aware Query Reformulation (§3.1.1)	Reformulates the original query to better retrieve reasoning-relevant context. This includes query decomposition (breaking complex queries into simpler ones) , reformulation (recasting ambiguous queries) , and expansion (enriching the query via CoT).	e.g., Collab-RAG (Xu et al., 2025b), DynQR (Anonymous, 2025), DeepRetrieval (Jiang et al., 2025)

Retrieval Strategy and Planning (§3.1.2)	Covers global retrieval guidance. This involves advance planning to generate a retrieval blueprint before execution or adaptive retrieval methods that predict whether and how to retrieve based on query complexity.	e.g., PAR-RAG (Zhang et al., 2025d), LPKG (Wang et al., 2024b), FIND (Jia et al., 2025)
Retrieval Model Enhancement (§3.1.3)	Enhances retrievers with reasoning. This is done by leveraging structured knowledge (like KGs with GNNs or symbolic rules) or integrating explicit reasoning (like CoT) with the query.	e.g., GNN-RAG (Mavromatis & Karypis, 2024), RuleRAG (Chen et al., 2024c)
Relevance Assessment & Filtering (§3.2.1)	Uses deeper reasoning to assess the relevance of retrieved fragments. This can involve using "assessor experts" to select faithful evidence or models to filter non-entailing passages.	e.g., SEER (Zhao et al., 2024c), M-RAG-R (Yoran et al., 2024)
Information Synthesis & Fusion (§3.2.2)	Fuses relevant snippets into a coherent evidence set after they are identified. Methods include aggregating sub-question answers or building a reasoning graph to evaluate and aggregate knowledge.	e.g., BeamAggR (Chu et al., 2024), DualRAG (Cheng et al., 2025), CRP-RAG (Xu et al., 2024)
Context-Aware Generation (§3.3.1)	Ensures outputs remain relevant and reduces noise. This includes selective-context utilization (pruning or re-weighting content) and reasoning path generation (building explicit logical chains).	e.g., Open-RAG (Islam et al., 2024), RARE (Wang et al., 2025d), Self-Reasoning (Xia et al., 2025b)
Grounded Generation Control (§3.3.2)	Introduces verification mechanisms to anchor outputs to retrieved evidence. This is done via fact verification , citation generation , and faithful reasoning (ensuring steps adhere to evidence).	e.g., RARR (Gao et al., 2023a), TRACE (Fang et al., 2024), AlignRAG (Wei et al., 2025b)
Knowledge Base (§4.1.1)	Retrieves from KBs storing arithmetic, commonsense, or logical knowledge. This can include formal lemmas for math , legal precedents , or code snippets.	e.g., Premise-Retrieval (Tao et al., 2025), ReaRAG (Lee et al., 2025), CBR-RAG (Wiratunga et al., 2024)
Web Retrieval (§4.1.2)	Accesses dynamic online content like web pages, news, or social media. It is used for fact-checking by verifying claims step-by-step or for QA by iteratively refining reasoning.	e.g., ALR^2 (Li et al., 2024d), RARE (Tran et al., 2024), Open-RAG (Islam et al., 2024)
Tool Using (§4.1.3)	Leverages external resources like calculators, libraries, or APIs to enhance reasoning interactively. This improves numerical accuracy and computational robustness.	e.g., TATU (Li et al., 2024g), TRICE (Qiao et al., 2024), Re-Invoke (Chen et al., 2024a)

Prior Experience (§4.2.1)	Retrieves past interactions or successful strategies stored in a model's internal memory. This includes leveraging past decisions for planning or recalling conversational histories for adaptive reasoning.	e.g., RAP (Kagaya et al., 2024), JARVIS-1 (Wang et al., 2024f), EM-LLM (Fountas et al., 2024)
Example or Training Data (§4.2.2)	Retrieves external examples from demonstrations or training data. This provides relevant exemplars to guide the model in emulating specific reasoning patterns.	e.g., MoD (Wang et al., 2024c), RE4 (Li et al., 2024c), UPRISE (Cheng et al., 2023)
Chain-based (§5.1.1)	Interleaves retrieval operations between the linear "step-by-step" reasoning of a Chain-of-Thought (CoT) to avoid error propagation. Methods can also add verification or filtering steps.	e.g., IRCOT (Trivedi et al., 2023), Rat (Wang et al., 2024g), CoV-RAG (He et al., 2024a), RAFT (Zhang et al., 2024a)
Tree-based (§5.1.2)	Explores multiple reasoning pathways. Tree-of-Thought (ToT) methods build a deterministic reasoning tree. Monte Carlo Tree Search (MCTS) methods use probabilistic tree search to dynamically prioritize exploration.	ToT: e.g., RATT (Zhang et al., 2025a), Tree of Clarifications (Kim et al., 2023) MCTS: e.g., AirRAG (Feng et al., 2025), MCTS-RAG (Hu et al., 2025)
Graph-based (§5.1.3)	Walk-on-Graph uses graph learning techniques (like GNNs) to retrieve and reason over graph-structured data. Think-on-Graph integrates graph structures into the LLM's reasoning loop, letting the LLM decide which node to explore next.	Walk-on-Graph: e.g., QA-GNN (Yasunaga et al., 2021) Think-on-Graph: e.g., ToG (Sun et al., 2024b), Graph-CoT (Jin et al., 2024)
Single-Agent (§5.2.1)	A single agent interweaves retrieval into its reasoning loop. This is achieved via Prompting (e.g., ReAct) , Supervised Fine-Tuning (SFT) , or Reinforcement Learning (RL).	Prompting: e.g., ReAct (Yao et al., 2023b) SFT: e.g., Toolformer (Schick et al., 2023) RL: e.g., Search-R1 (Jin et al., 2025)
Multi-Agent (§5.2.2)	Uses multiple agents for collaboration. Decentralized systems use specialized agents that work together. Centralized systems use a hierarchical (e.g., manager-worker) pattern for task decomposition.	Decentralized: e.g., M-RAG (Wang et al., 2024) Centralized: e.g., HM-RAG (Liu et al., 2025), Chain of Agents (Zhang et al., 2024c)

The survey highlights a clear evolution from static retrieval pipelines toward dynamic, agent-based RAG systems with deep reasoning capabilities. It identifies key challenges such as efficiency, evaluation, and controllability, while outlining future research directions that aim to unify reasoning, retrieval, and agent learning into a coherent framework for next-generation LLM systems.

5.3 RAG-RL: Advancing Retrieval Augmented Generation via RL and Curriculum Learning

Link to paper: https://arxiv.org/abs/2503.12759 [3]

6 Implementation: LangChain, Text-to-speech, Speech-to-text

6.1 Building an AI Agent Chatbot with LangChain

We implement AI agent chatbot using the LangChain framework, designed to support a music-related website through Retrieval-Augmented Generation (RAG) combined with speech-based interaction. LangChain enables the construction of agentic systems by integrating large language models (LLMs) with external tools, memory, and retrieval mechanisms, allowing the chatbot to reason, decide actions, and iteratively solve user queries.

An **AI agent** in LangChain is defined as a system that combines a language model with a set of tools, enabling it to select and invoke appropriate tools based on the task context. Tools act as functional interfaces that extend the model's capabilities beyond pure text generation, such as searching documents or querying the web.

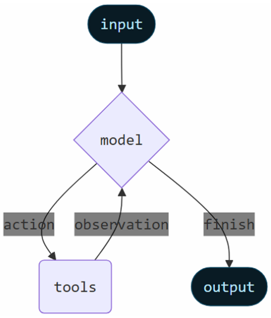


Figure 8: AI Agent

6.1.1 RAG Agent Architecture

The chatbot is built around a RAG agent architecture consisting of three main stages: indexing, retrieval, and response generation. During the indexing phase, nearly 100 Wikipedia articles related to Music are loaded using LangChain's `WikipediaLoader`. These documents are embedded using the `gemini-embedding-001` model and stored persistently in a `Chroma` vector database, enabling efficient semantic similarity search.

6.1.2 Agent Tools

The agent is equipped with multiple tools to enhance its reasoning and information access:

- **Context Retrieval Tool**: Retrieves the top-k (with $k = 10$) most relevant documents from the Chroma vector store based on semantic similarity.
- **Web Search Tool**: Uses DuckDuckGo to fetch real-time search results, including URLs, titles, and snippets, enabling access to up-to-date information.
- **Web Loader Tool**: Employs LangChain's `WebBaseLoader` to extract and process raw HTML content from selected web pages.

These tools are wrapped and exposed to the agent, allowing it to dynamically decide whether to rely on internal knowledge, retrieved documents, or external web sources.

6.1.3 Agent Configuration and Memory

The agent is initialized with a system prompt that defines its role and behavior. The conversational backbone uses the `gemini-2.5-flash` chat model for response generation. To maintain contextual coherence, short-term memory is incorporated, enabling the agent to remember and reference previous turns within a single conversation thread.

6.2 Text-to-speech

6.3 Speech-to-text

7 Evaluation

We need to evaluate how effective our RAG system is. We will be rating the RAG feature, not the web search tool here. As discussed in the previous section, the RAG vector store is taken from 76 Wikipedia articles about Music. We will also measure the time it take to retrieve the relevance context from the database.

We use 2 type of evaluation score:

1. **Cosine Similarity (0-1)**: Semantic similarity between query embeddings and document embeddings
2. **LLM-as-a-Judge (1-5 scale)**:
    - 5 = Highly relevant, directly answers query
    - 4 = Relevant, provides useful information
    - 3 = Somewhat relevant, tangentially related
    - 2 = Minimally relevant
    - 1 = Not relevant

We use 10 text querries: Basic factual queries (1-2), Temporal queries (3-5), Event-based queries (6-7), Complex analytical queries (8-10)

1. What are the most common musical instruments?
2. Explain the concept of harmony in music
3. How has music evolved over the centuries?
4. What are the characteristics of music in the Renaissance period?
5. When did electronic music emerge?
6. What impact did the invention of recording technology have on music?
7. How did jazz influence modern music genres?
8. Compare Western and Eastern musical traditions
9. What is the relationship between rhythm and cultural identity in music?
10. How do musical scales affect emotional perception?

For each of the 10 text querries, we first retrieve the top 5 relevant documents from the vector store. Then we use the 2 method above to evaluate the relevant score of the retrieved docs.

1. For method 1, we calculate the embedding of the query and of each document, then we calculate the cosine similarity between the embedding of query with the embedding of each doc.
2. For method 2, we feed a prompt to the LLM to ask it to give a score based on how relevant the document is to the query. The model used is `gemini-2.5-flash`.

```python
for i, doc in enumerate(retrieved_docs):
    prompt = f"""Rate the relevance of the following document to the query on a scale of 1-5.

Query: {query}

Document Content: {doc.page_content[:500]}...

Scoring scale:
5 - Highly relevant, directly answers the query
4 - Relevant, provides useful information
3 - Somewhat relevant, tangentially related
2 - Minimally relevant, barely related
1 - Not relevant, unrelated to query

Respond with ONLY the numeric score (1-5)."""

    try:
        response = model.invoke(prompt)
        score = int(response.content.strip())
        scores.append(min(max(score, 1), 5))  # Ensure score is between 1-5
    except:
        scores.append(3)  # Default to neutral if parsing fails
```

Listing 1: LLM-as-a-judge

Here is the result:

```
============================================================
Query 1: What are the most common musical instruments?
============================================================

Metrics:
  Retrieval time: 1.732s
  Avg similarity: 0.697
  Avg relevance: 2.60/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.715
    Relevance: 3/5
    Source: https://en.wikipedia.org/wiki/Musical_instrument
    Content: as the only system that applies to any culture and, more importantly, provides the only possible classification for each instrument. The most common c...

  Document 2:
    Similarity: 0.703
    Relevance: 4/5
    Source: https://en.wikipedia.org/wiki/Percussion_instrument
    Content: === By prevalence in common knowledge ===
It is difficult to define what is common knowledge but there are instruments percussionists and composers us...

  Document 3:
    Similarity: 0.697
    Relevance: 2/5
    Source: https://en.wikipedia.org/wiki/Musical_instrument
    Content: There are many different methods of classifying musical instruments. Various methods examine aspects such as the physical properties of the
    instrument...

============================================================
Query 2: Explain the concept of harmony in music
============================================================

Metrics:
  Retrieval time: 0.483s
  Avg similarity: 0.755
  Avg relevance: 5.00/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.773
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: Harmony refers to the "vertical" sounds of pitches in music, which means pitches that are played or sung together at the same time creates a chord. Us...

  Document 2:
    Similarity: 0.764
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Music_theory
    Content: In music, harmony is the use of simultaneous pitches (tones, notes), or chords. The study of harmony involves chords and their construction and chord ...

  Document 3:
    Similarity: 0.751
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: === Harmony ===...

============================================================
Query 3: How has music evolved over the centuries?
============================================================

Metrics:
  Retrieval time: 0.465s
  Avg similarity: 0.720
  Avg relevance: 4.20/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.731
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Popular_music
    Content: There are multiple possible explanations for many of these changes. One reason for the brevity of songs in the past was the physical capability of rec...

  Document 2:
    Similarity: 0.730
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Classical_music
    Content: By the 20th century, stylistic unification gradually dissipated while the prominence of popular music greatly increased. Many composers actively avoid...

  Document 3:
    Similarity: 0.718
    Relevance: 3/5
    Source: https://en.wikipedia.org/wiki/Popular_music
    Content: In addition to many changes in specific sounds and technologies used, there has been a shift in the content and key elements of popular music since th...

============================================================
Query 4: What are the characteristics of music in the Renaissance period?
============================================================

Metrics:
  Retrieval time: 0.479s
  Avg similarity: 0.724
  Avg relevance: 4.40/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.744
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: Renaissance music (c.1400 to 1600) was more focused on secular themes, such as courtly love. Around 1450, the printing press was invented, which made...

  Document 2:
    Similarity: 0.734
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Classical_music
    Content: ==== Renaissance ====

The musical Renaissance era lasted from 1400 to 1600. It was characterized by greater use of instrumentation, multiple interwea...

  Document 3:
    Similarity: 0.722
    Relevance: 4/5
    Source: https://en.wikipedia.org/wiki/Baroque_music
    Content: tritone, perceived as an unstable interval, to create dissonance (it was used in the dominant seventh chord and the diminished chord). An interest in ...

============================================================
Query 5: When did electronic music emerge?
============================================================

Metrics:
  Retrieval time: 0.571s
  Avg similarity: 0.728
  Avg relevance: 4.60/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.741
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Electronic_music
    Content: The first electronic musical devices were developed at the end of the 19th century. During the 1920s and 1930s, some electronic instruments were intro...

  Document 2:
    Similarity: 0.740
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Electronic_music
    Content: During the 1960s, digital computer music was pioneered, innovation in live electronics took place, and Japanese electronic musical instruments began t...

  Document 3:
    Similarity: 0.724
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Electronic_music
    Content: === United States ===
In the United States, electronic music was being created as early as 1939, when John Cage published Imaginary Landscape, No. 1, ...

============================================================
Query 6: What impact did the invention of recording technology have on music?
============================================================

Metrics:
  Retrieval time: 0.472s
  Avg similarity: 0.708
  Avg relevance: 5.00/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.715
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: distributed. The introduction of the multitrack recording system had a major influence on rock music, because it could do more than record a band’s pe...

  Document 2:
    Similarity: 0.712
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Popular_music
    Content: In the 1950s and 1960s, the new invention of television began to play an increasingly important role in disseminating new popular music. Variety shows...

  Document 3:
    Similarity: 0.709
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: In the 19th century, a key way new compositions became known to the public was by the sales of sheet music, which middle class amateur music lovers wo...

============================================================
Query 7: How did jazz influence modern music genres?
============================================================

Metrics:
  Retrieval time: 0.539s
  Avg similarity: 0.705
  Avg relevance: 3.60/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.719
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Rock_music
    Content: fusion began to take its audience, but acts like Steely Dan, Frank Zappa and Joni Mitchell recorded significant jazz-influenced albums in this period,...

  Document 2:
    Similarity: 0.712
    Relevance: 3/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: Jazz evolved and became an important genre of music over the course of the 20th century, and during the second half, rock music did the same. Jazz is ...

  Document 3:
    Similarity: 0.702
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Rock_music
    Content: British acts to emerge in the same period from the blues scene, to make use of the tonal and improvisational aspects of jazz, included Nucleus and the...

============================================================
Query 8: Compare Western and Eastern musical traditions
============================================================

Metrics:
  Retrieval time: 0.499s
  Avg similarity: 0.720
  Avg relevance: 4.40/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.734
    Relevance: 4/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: In the West, much of the history of music that is taught deals with the Western civilization's art music, known as classical music. The history of mus...

  Document 2:
    Similarity: 0.720
    Relevance: 4/5
    Source: https://en.wikipedia.org/wiki/Classical_music
    Content: Classical music generally refers to the art music of the Western world, considered to be distinct from Western folk music or popular music traditions....

  Document 3:
    Similarity: 0.715
    Relevance: 4/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: === Asian cultures ===

Asian music covers a swath of music cultures surveyed in the articles on Arabia, Central Asia, East Asia, South Asia, and Sout...

============================================================
Query 9: What is the relationship between rhythm and cultural identity in music?
============================================================

Metrics:
  Retrieval time: 0.477s
  Avg similarity: 0.696
  Avg relevance: 3.20/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.709
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Tempo
    Content: song (although this would be less likely with an experienced bandleader). Differences in tempo and its interpretation can differ between cultures, as ...

  Document 2:
    Similarity: 0.698
    Relevance: 3/5
    Source: https://en.wikipedia.org/wiki/Tempo
    Content: This context-dependent perception of tempo and rhythm is explained by the principle of correlative perception, according to which data are perceived i...

  Document 3:
    Similarity: 0.695
    Relevance: 2/5
    Source: https://en.wikipedia.org/wiki/Music
    Content: === Rhythm ===...

============================================================
Query 10: How do musical scales affect emotional perception?
============================================================

Metrics:
  Retrieval time: 0.451s
  Avg similarity: 0.705
  Avg relevance: 3.00/5

Top 3 Retrieved Documents:

  Document 1:
    Similarity: 0.709
    Relevance: 5/5
    Source: https://en.wikipedia.org/wiki/Music_theory
    Content: The interrelationship of the keys most commonly used in Western tonal music is conveniently shown by the circle of fifths. Unique key signatures are a...

  Document 2:
    Similarity: 0.708
    Relevance: 4/5
    Source: https://en.wikipedia.org/wiki/Scale_(music)
    Content: Tetratonic (4 notes), tritonic (3 notes), and ditonic (2 notes): generally limited to prehistoric ("primitive") music
Scales may also be described by ...

  Document 3:
    Similarity: 0.704
    Relevance: 2/5
    Source: https://en.wikipedia.org/wiki/Soundtrack
    Content: plot anticipations, and moral judgement of the characters.
    Furthermore, eyetracking and pupillometry studies found that film music is
    able to influenc...

============================================================
```

Listing 2: RAG score result

Overall, we can see that the time it take to retrieve the context is acceptable (less than 1 second for most cases). The average similarity score is around 0.7-0.75; and the average LLM judge rating varied more, around 2.6 to 5.0. LLM judge score can be pretty inconsistance and biased, or even inaccurate, so we use extra manual inspection to check the relevance of the retrived context to the query; but generally, the LLM can be quite good at deciding which infomation is important to answer the query.

In conclusion, we can say that with our vector database of 76 Wiki articles is quite sufficient for our RAG system, and RAG did a good job at retrieving the neccesary context for the queries.

8 Conclusion

References

[1] Yunfan Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey.* Preprint. 2023. DOI: 10.48550/arXiv.2312.10997. arXiv: 2312.10997 [cs.CL]. URL: https://arxiv.org/abs/2312.10997.

[2] Google AI for Developers. *Google AI Studio.* Accessed: 2026-01-08. Google. 2025. URL: https://ai.google.dev/aistudio.

[3] Jerry Huang et al. *RAG-RL: Advancing Retrieval-Augmented Generation via RL and Curriculum Learning.* Preprint. 2025. DOI: 10.48550/arXiv.2503.12759. arXiv: 2503.12759 [cs.CL]. URL: https://arxiv.org/abs/2503.12759.

[4] LangChain Documentation. *Build a RAG agent with LangChain.* Accessed: 2026-01-05. LangChain. 2025. URL: https://docs.langchain.com/oss/python/langchain/rag.

[5] Yangning Li et al. *Towards Agentic RAG with Deep Reasoning: A Survey of RAG-Reasoning Systems in LLMs.* Preprint. 2025. DOI: 10.48550/arXiv.2507.09477. arXiv: 2507.09477 [cs.CL]. URL: https://arxiv.org/abs/2507.09477.

[6] OpenAI. *Introducing ChatGPT.* Accessed: 2026-01-08. OpenAI. 2022. URL: https://openai.com/index/chatgpt/.