FRI Natural language processing course 2025
Project report

University *of Ljubljana*
Faculty *of Computer and Information Science*

Conversational Agent with Retrieval-Augmented Generation

Matej Belšak, Gorazd Gorup, Luka Bajić

Abstract

In this report, we developed a set of RAG solutions and performed a thorough evaluation. We used two LLMs, DeepSeek and Qwen3, to compare the behavior of RAG with and without reasoning, respectively. We built two methods for information retrieval and injection, one based on prompt analysis and nondiscriminative prompt injection, and one based on function calling at the LLM's discretion. To perform more robust testing, we evaluated all variations of our RAG systems on a prepared question set, using two different-sized LLMs-as-judge – GPT-4.1.-mini and 14-billion-parameter Qwen model – and our own subjective observations. We found that both judges performed similarly on some metrics, with GPT-4.1.-mini performing better on others. The results show that Qwen with the function-calling RAG system outperforms all other variants.

Keywords
Conversational agent, Retrieval-Augmented Generation

Advisors: Aleš Žagar

Introduction

While Large Language Models (LLMs) have evolved considerably and now produce convincing replies, they have inherent limitations. They rely on training data consisting of documents from the past and may not possess knowledge of current events and developments. Due to differences and properties of training datasets, they may not contain specific domain knowledge, failing to answer certain prompts or outright hallucinating. The latter could prove especially disastrous in dedicated chatbots, for example for legal guidance or health care [1].

One possibility would be to retrain the chat model in intervals to try to keep it up to date, but that would still produce periods where the information is missing from the LLM or is outdated. This approach, while completely time- and energy-inefficient, would also not guard against hallucinations. To solve these issues, Retrieval Augmented Generation (RAG) is used to provide the missing knowledge to the LLM. RAG employs different techniques to retrieve information from external sources based on the user's prompt and, through prompt augmentation, feed the LLM sufficient information to provide an informative and factually correct answer. In the survey by Gao *et al.*, [2], multiple approaches to RAG are presented, highlighting three architectures: naive RAG, which analyzes the user's prompt, retrieves the required information, and appends it, letting the LLM do the rest; advanced RAG, which employs pre-retrieval and post-retrieval modifications to the prompt to make it more suitable for information retrieval and subsequent interpretation by LLM; lastly, modular RAG combines multiple approaches, using iterative prompt enhancement, ranking, fusion, etc.

For document summarization, LLMs, statistical models, graph-based models, and other approaches are used to extract the most important information from text. Zhang *et al.*, [3] present multiple solutions, noting that LLMs, while consuming more resources, tend to be more coherent and precise in their summarization if trained correctly.

In this paper, we focus on RAG methods and their use in chatbots. To that end, we design a conversational agent operating on knowledge about different art and media. Specifically, the agent is to suggest and converse about films and other related media based on the user's prompts and preferences. Our contributions are:

- Implementation of a conversational agent using two different pretrained models: DeepSeek-R1 [4] and Qwen 3 [5].
- Implementation of two RAG techniques, a primitive and advanced one, with capabilities of retrieving data from various film-related databases and web sources.
- Evaluation of conversational agents regarding the model used and the RAG technique. Agents are evaluated

against the baseline RAG-less chatbots. Human and LLM judges are used to score responses, and different metrics are analyzed.

Methods

We designed two RAG pipelines, as presented in fig. 1. The naive RAG pipeline used Natural Language Processing (NLP) methods to extract the important feature information from prompts via lemmatization, stop-word removal, and entity detection. Prompt information was used to determine movie titles and names of people to query for in our selected databases. That information was then appended to the prompt and fed into an LLM.

A more advanced version of our RAG system utilized various components, but we ultimately chose to employ function-calling functionality that is present in some LLMs. Based on the prompt, the LLM decides on the most appropriate information retrieval function to call and informs the RAG system, which then retrieves that information. The information is then inserted into the prompt, which is fed into the LLM.

Function Calling

In some cases, we leveraged the function calling capabilities of our models. While DeepSeek hints at the possibility of using this feature [6], their proposed prompt templates do not support it. We instead turned to Qwen's function calling capabilities, which proved to be sufficient.

In function calling, the model is presented with the user prompt and structured information about programmatic functions it may call to fullfil the user's request. The LLM is tought to return a similarly structured set of instructions on which functions to call, and with what parameters.

Information Retrieval

To provide a sufficient level of information related to movies and similar media, we selected the following sources:

- **The Movie Database (TMDB).** This is an open database of movies, TV shows, and people associated with them. They provide a free-to-use API for various types of requests – release dates, cast, crew, similar films ...
- **Letterboxd.** This is a movie-themed social network to log watched films, provide reviews, and engage in movie-related discussions. We chose to include it to give the model a better understanding of how people perceive certain media.
- **JustWatch.** This website tracks the availability of media on digital and streaming platforms.
- **Wikipedia.** We used it to retrieve information such as film plots, summaries, etc.

In our naive RAG implementation, we retrieved data from all four sources. This would amount to over 250,000 characters of context.

For our advanced version, we prepared nine custom scraping functions that would equip the model with essential information about movie titles and film-related people. These functions had to be specifically annotated, and various versions of descriptions were used to elicit a proper function-calling response from the model.

RAG

For the naive RAG system, we first analyzed the user prompt and gathered all relevant entities, consisting of film titles and names of people. We used a Roberta-like `spacy` model [1]. We performed unfiltered knowledge injection, gathering all relevant documents based on prompt entities, structuring them in JSON format, and passing them along with user query to the LLM.

For the advanced RAG system, we tried several versions before basing it on function calling. Our earlier attempts performed the entity extraction from prompt, retrieval of documents from all sources based on those entities, and content curation based on statistical methods and user prompt. For example, we treated all sentences in the retrieved data as documents, performing TF-IDF sorting and using only the top N most relevant sentences. Unfortunately, results of these extractions accounted only for literal appearance of words from the prompt in documents, instead of accounting for larger context, synonyms, etc.

To address these shortcomings, we turned to function calling, letting the LLM decide what information to retrieve. In the system prompt, we also appended additional information about the current date for better context formulation.

With function calling, we also had to thoroughly describe all available functions to the LLM, which proved challenging, as the model sometimes failed to infer the correct connection between the user's input and the function descriptions.

Prompt formulation

We structured our prompts to consist of three roles: system, user, and assistant. We put system prompt first, followed by the user prompt, and finally by an assistant tag to signal the LLM to generate the answer, not to continue the user prompt.

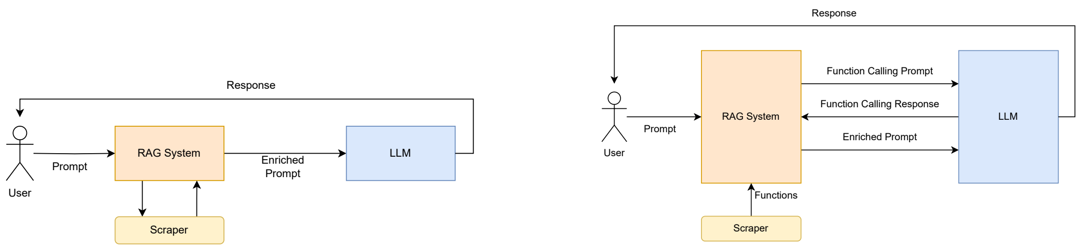
Our system prompt is shown in fig. 2. The user prompt consists of the user's initial query and the retrieved information, enclosed in `<data>` tags, in that order.

A key challenge during this stage was ensuring that the LLM relied solely on the provided information and avoided generating unsupported content. Despite clear instructions, it was often difficult to prevent the model from hallucinating or making assumptions beyond the data enclosed in the `<data>` tags.

Memory and chat

To extend a question-answering model into a conversational chatbot, we kept the information retrieval pipeline unchanged and implemented a simple answer caching mechanism to

[1] https://huggingface.co/spacy/en_core_web_sm



(a) A naive RAG system. **(b)** An advanced RAG system.

Figure 1. Our proposed RAG pipelines for evaluation. The naive RAG variant preprocesses the prompt, extracts meaningful information from it, retrieves the data based on that, and enhances the prompt with that raw data. The advanced variant employs LLM function-calling capabilities to decide what data to scrape and enhances the prompt only with necessary data.

> You are an AI chatbot, assisting user with anything related to movies. [Today's date is `{date}`.] You may only use information provided to you inside the `<data>` tags.

Figure 2. Out prompt for RAG usage. The text in square brackets was only used in advanced RAG, and the text in braces represents variables.

preserve conversational context. Specifically, we stored the last *n* user queries and chatbot responses using the First-In-First-Out (FIFO) method and included them in the prompt provided to the LLM. Due to input token limitations, the value of *n* had to remain relatively small in our experiments to ensure the model could process the full prompt effectively. Otherwise, there was a risk of token truncation and generation errors.

Evaluation methodology

Since we are working with relatively open-ended questions, there is a lack of objective ground truth to evaluate against. Therefore, we construct a set of 50 domain-relevant questions and use manually checked answers from the commercial ChatGPT model as ground truth for metric computations. To avoid overfitting a model to a specific set of expected outputs, we included multiple types of questions into our test set, including yes/no questions, fact checking, list retrieval, and summarization. All queries are zero-shot.

LLM-based evaluation

Standard evaluation metrics such as BLEU and ROUGE are ill-adjusted to our test set because they penalize diverse answers, which could still be correct in the context of open-ended questions. To bypass this limitation, we employ the DeepEval framework [7] to obtain more relevant metrics based on context, query and response:

- **Correctness**: Measures how accurate the response is compared to the ground truth.
- **Clarity**: Evaluates how clear and understandable the response is for the user. (We do not expect to improve this metric with RAG, we only utilize it to ensure that our modifications do not degrade the original model's inherent capabilities).
- **Answer Relevancy**: Assesses how well the response addresses the user's input and stays on topic.
- **Faithfulness**: Measures whether the response is factually consistent with the provided context, without introducing unsupported claims.
- **Contextual Precision**: Evaluates whether relevant pieces of context are ranked higher than irrelevant ones for the given user's input.
- **Contextual Recall**: Measures how well the retrieved context covers the information needed to produce the ground truth response.
- **Contextual Relevancy**: Assesses the overall relevance and usefulness of the retrieved context for answering the user's input.

The correctness, clarity, and answer relevancy show the LLM's focus on the retrieved information regarding the ground truth and the user's query. Faithfulness measures the LLM's capability to properly use the retrieved information. Precision, recall, and contextual relevancy all present the performance of the retrieval system. The models were judged on a scale from 0 to 1.

Results

Since we were limited by hardware capabilities during our research, we focused on models with quantized or distilled variants. We ultimately decided on a Qwen3 [5] [2] model as our non-reasoning LLM, and a distilled Llama variant of DeepSeek R1 [4] [3] as our reasoning LLM. Both models were loaded from HuggingFace and were run through the `transformers` API.

Both models were 8-billion-parameters versions. We chose DeepSeek as it is a novel set of models with positive

[2] https://huggingface.co/Qwen/Qwen3-8B
[3] https://huggingface.co/deepseek-ai/DeepSeek-R1

model	correctness	clarity	answer relevancy	faithfulness	contextual precision	contextual recall	contextual relevancy
deepseek-baseline	0.1280	0.7780	0.7657	—	—	—	—
deepseek-naive	0.1720	0.7800	0.6846	0.9310	0.1600	0.5097	0.4238
deepseek-advanced	0.2440	0.6920	0.8055	**0.9397**	**1.0000**	**1.0000**	**0.5580**
qwen-baseline	0.2950	**0.8950**	0.7283	—	—	—	—
qwen-naive	0.2780	0.8425	**0.8510**	0.8530	0.9500	0.8950	0.3370
qwen-advanced	**0.4400**	0.8200	0.7411	0.9280	0.2000	0.4000	0.1774

Table 1. Performance comparison as evaluated by a 14B-parameter Qwen model with DeepEval framework.

model	correctness	clarity	answer relevancy	faithfulness	contextual precision	contextual recall	contextual relevancy
deepseek-baseline	0.3185	0.7492	0.6172	—	—	—	—
deepseek-naive	0.3341	0.8144	0.7207	0.7561	0.3400	0.3201	0.4362
deepseek-advanced	0.3341	0.7841	0.8097	0.9301	**1.0000**	0.6094	0.5681
qwen-baseline	0.4521	0.8102	**0.8794**	—	—	—	—
qwen-naive	0.4512	0.7799	0.8001	0.8323	0.3400	0.2644	0.5164
qwen-advanced	**0.5234**	**0.8338**	0.6804	**0.9794**	**1.0000**	**0.6110**	**0.6836**

Table 2. Performance comparison as evaluated by GPT-4.1-mini model with DeepEval framework.

benchmarking results [4]. For each of these, we run experiments on three different sets of parameters:

- without RAG - out-of-the-box model with no modifications (baseline)
- with naive RAG - all documents are retrieved and appended to the context with no consideration for token limitation (data is truncated)
- advanced RAG - documents are selectively retrieved and processed before being appended to the context

Results obtained with the Qwen3-14B model are shown in Table 1. Metrics related to context are not available for baseline models because their context is an empty set. Models with advanced RAG can choose not to retrieve documents, depending on the user query, so the context-related metrics are only computed when at least one document was retrieved.

Because the model used for evaluation is not drastically larger than the model used for response generation, it is sensible to repeat the same experiments with an even larger LLM to obtain more reliable results. We opted for GPT-4.1-mini. Results are shown in Table 2. Due to computational complexity, we compute the metrics on the first 25 questions in the test set and report the average values.

Conversational chatbot

In advanced RAG, we observed that the memory mechanism prevented the chatbot from being confused about queries referencing previous ones, managing to remember crucial data for function calling. Sometimes, the LLM even decided against retrieving information in queries where it otherwise would because previous questions and answers provided enough information or context to formulate a sufficient response.

Examples of Qwen chatbot results are shown in the Appendix.

Discussion

As seen in Table 1, Qwen judge scores DeepSeek with advanced RAG quite high on the information retrieval metrics, while preferring Qwen in LLM answer generation based on ground truth and user prompt. Qwen with advanced RAG is considered to be the most in line with ground truth answers. Interestingly, Qwen judge considers even the best model to perform poorly on more than half the questions regarding correctness.

Table 2 shows that GPT tends to evaluate the results more favorably than Qwen.

Notably, Qwen with advanced RAG system performs best in most metrics. Specifically, it generally generates responses relevant to the user's question. When given context, it follows the provided information most faithfully. It performs decently when compared to the ground truth, passing over 50% of the tests.

Our advanced RAG system shows that our retrieved data is mostly aligned with what is expected, although we sometimes present the LLM with irrelevant information. This can be attributed to the fact that our retrieval does not perform granular information extraction, instead relying on a function-calling model to pick specific (short) documents of information to retrieve.

Baseline Qwen returns a more relevant answer, which we could attribute to the fact that it has the most freedom when answering, not tied to the context, which could lead the answer to not directly align with what the query is asking or cause the LLM to populate the answer with unnecessary details.

In addition to evaluating RAG methods, we also explored how large a model needs to be to effectively serve as an LLM-as-judge for smaller models. Based on both metric results and manual inspection of model outputs against ground truth answers, we observed that GPT-4.1-mini was better at fairly and accurately evaluating responses. Qwen3-14B often failed to do so reliably, which is why we place more trust in GPT-based evaluations. Our results also confirm that the advanced RAG approach outperforms the naive one across models, with Qwen generally performing better than DeepSeek.

References

[1] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

[2] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[3] Yang Zhang, Hanlei Jin, Dan Meng, Jun Wang, and Jinghua Tan. A comprehensive survey on process-oriented automatic text summarization with exploration of LLM-based methods, 2025.

[4] DeepSeek-AI et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025.

[5] Qwen Team. Qwen3, April 2025.

[6] DeepSeek. Function Calling — DeepSeek API Docs — api-docs.deepseek.com. https://api-docs.deepseek.com/guides/function_calling/. [Accessed 29-05-2025].

[7] Jeffrey Ip and Kritin Vongthongsri. deepeval, May 2025.

Appendix

Conversation Example with Advanced Qwen

What is the movie Until Dawn about?

Until Dawn is a 2025 American **survival horror film** directed by **David F. Sandberg** and written by **Gary Dauberman** and **Blair Butler**. Based on the **2015 video game** *Until Dawn* developed by **Supermassive Games** for PlayStation, the film adapts the game's eerie atmosphere and gameplay mechanics—such as the **time-loop structure** and **branching narratives**—into a **standalone storyline** within the same fictional universe.

Key Details About Until Dawn

`[... (omitted due to length)]`

Who directed that movie?

The movie **Until Dawn (2025)** was directed by **David F. Sandberg**, a Swedish filmmaker best known for his work on the *Annabelle* and *Lights Out* franchises. He brought his signature style of suspenseful horror and atmospheric tension to this adaptation of the popular video game *Until Dawn*.
`[... (omitted due to length)]`

Who are the main actors?

Here are the **main cast members** of *Until Dawn (2025)* along with their roles:

Main Cast:

Actor Name	Role
Ella Rubin	**Clover Paul**
Michael Cimino	**Max**
Odessa A'zion	**Nina Riley**
Ji-young Yoo	**Megan**
Belmont Cameli	**Abe**
Maia Mitchell	**Melanie Paul**
Peter Stormare	**Dr. Alan Hill**

`[... (omitted due to length)]`
Would you like information about any particular actor or role from the film?