FRI Natural language processing course 2024/2025
Project report

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

Project 1: Conversational Agent with Retrieval-Augmented Generation

Anja Pijak, Lara Anžur, and Dimitrije Stefanović

Abstract

In this paper, we present a conversational agent that leverages Retrieval-Augmented Generation (RAG) and web scraping techniques to provide accurate, up-to-date responses focused on academic research. Our system combines multiple academic paper sources, with dynamic web scraping capabilities to create a comprehensive knowledge base. The agent uses vector similarity search and can handle both general queries and website-specific information retrieval. We demonstrate how this approach addresses common limitations of traditional language models, such as outdated information and hallucinations, while providing targeted academic research assistance.

Keywords

Natural Language Processing, Retrieval-Augmented Generation, Web Scraping, Large Language Models, Academic Papers, Academic Research, Ai in Academia

Advisors: Aleš Žagar

Introduction

It is widely known that the usefulness of the many Large Language Models are often offset by the colloquial hallucinations regularly found in the answers to specific questions. Looking to better put the focus on the information retrieval in answers of the AI agents, many of the models in use today make use of the Retrieval-Augmented Generation (RAG) technology.

The motivation for writing this paper and the research on the possibilities of integrating RAG technology and web scraping in models, comes from the academic experiences and use of Large Language Models and similar AI technologies in such an environment.

Our research and view is of that, that many such existing agents are too broad in area of search and simply lack the proper guiding, algorithms and fine-tuning that is needed for them to be used as a help in research in any of the academic spaces.

Many of the AI agents in use today do utilize web scraping and use the relevant information scraped from the web to fuel and better inform the answers that are being processed by them. The function of the RAG technology is a far better researched topic and is in active use today in many effective and vastly popular agents; however, our aim is to analyze a way for an agent to be far more relevant for research and study of academic papers, able to effectively extract knowledge and broaden the extent of the groundwork for further study.

Our goal is to find a way for a simple agent to be more focused and, with that, far more precise in its findings for a specific topic, being able to speed up research and development to bring more relevant research papers to far more people.

We will view how a simple use of pretrained models, combined with the use of external data, acquired through web scraping, can help us understand the possibilities of this view of the NLP landscape. This agent will aim to be far more focused, finding direct links to actual submitted papers without being outdated in its information

Related work

The development of conversational agents with Retrieval-Augmented Generation (RAG) has been a growing area of research, with the aim of improving response accuracy by integrating external knowledge sources. This section reviews existing studies that have contributed to this field.

From Gao et al. [1] we have substantial examples of why RAG is important and how it helps AI be more accurate and reliable. The authors propose a novel RAG model that combines a large language model with a retrieval mechanism, allowing the model to generate responses based on both pre-trained knowledge and external sources. This approach significantly improves the accuracy of the responses, making the model more reliable and informative.

For this project, we will, however, focus on a type of RAG that incorporates web-scraped data to address the problem of outdated information. The knowledge embedded in large language models tends to become outdated over time [2], making it crucial to retrieve and integrate external sources dynamically.

Pokhrel et al. [3] explore the integration of AI and web scraping in RAG. Their approach combines web scraping, vectorization, and semantic search, allowing users to input a specific website and then ask questions exclusively based on its content. While this method effectively retrieves domain-specific information, it requires the user to specify a website beforehand. In our work, we wish to build upon this approach by enabling both targeted and general prompts. If a user requests information from a specific website, our chat-bot will focus on that source. However, if no specific website is provided, the system will still perform web scraping in the background, integrating newly retrieved knowledge with the model's pre-trained information. This ensures that responses remain accurate and up to date while maintaining flexibility for both broad and focused queries.

The concept of leveraging web-scraped data through RAG was explored by Kanataria et al. [4], where authors proposed a novel framework that integrates RAG techniques with large language models (LLMs) like Llama 3, Mistral, and Gemini, enhancing the process of data retrieval and summarization from online sources. This framework combines web scraping, embedding models, and FAISS for efficient indexing, ensuring the delivery of real-time, personalized information. Our goal is to combine this approach with the one suggested by Pokhrel et al. [3], allowing the chatbot to dynamically retrieve and integrate information from any user-provided website, as well as from general web scraping.

Methods

In the sample work of our proof-of-concept model setup, the focus was mainly on the various resources and ways of implementing them. We've managed to group together and take use of many knowledge bases of academic research and compile the into the sample RAG pipeline that manages to access these papers in a keyword fashion and give answers enriched with these resources.

Papers, resources and fetching

In implementing many of these papers, that our python script fetches, they are then converted into vectors, and chosen based on similarity, using the FAISS library. There are many ways of obtaining academic papers that have been found and in turn implement in the test of the capabilities of the pipeline.

Many of the papers are possible to be obtained and have public API interfaces that gives much leeway in what and in what way we search.

Popular and massive academic libraries such as Arxiv and Semantic Scholar already have available python libraries allowing us the seamlessly implement them and their resources into our project. While other sites such as GoogleScholar and CORE have publicly open API interfaces still making it possible to quickly fetch the data.

While these resources are in fact readily available to us, some of the websites containing them were only available to be accessed through web scraping. Integrating the Playwright library and some simple web scraping methods, makes it possible to also implement these papers into the vectorization of this data.

Models, embeddings and similarity

While the baseline implementation only included simple and one model testing, it is important to highlight the different methods that are important in this situation and context.

The embedding model is also an aspect not that much highlighted in our testing case, but something to be researched upon further, given the importance of our RAG data

Our implementation also scores different specific hyperparameters that may be important to find the difference between.

For example the amount of the papers being fetched from remote resources are currently limited, to highlight rapid development of the model pipeline. While the amount currently is very much useful to be low, it is an important resource for our model that should be analyzed further and it's importance to the final result.

Similarly to the amount of papers being fetched, the k parameter in the FAISS vectorization is something that could massively impact the result and effectiveness of the data being implement to our answers

Analysis

For the simple demonstration of the capabilities of running a model that incorporates RAG techniques with scientific papers, we have made a short testing suite based on some sample questions one could maybe pose when researching for a topic in an academic setting. Because the baseline implementation lacks proper testing and fine-tuning, while pinning the results against proper, in-use, online AI agents, we've appended to the query the following: "Answer the question shortly". In this way, the agents will respond in shorter answers more uniform to our baseline implementation, given that in the opposite case, we get rather large and formatted answers which do not quite align with our results for our baseline implementation.

We've constructed queries, each with some keywords attached to them, that our agent could search for proper context, and asked it the question in connection to the keywords.

The testing agents, we're simply given the question, as stated above.

In testing it has shown that ChatGPT has given quite more consistent results and to the point answers than the agent

Claude, while our implementation was quite similar to it in that regard.

The responses our agent pipeline has shown have exhibited similarities in how a student could write in his own seminar paper, which aligns with our goals of this proof of concept. The writing is formal and corresponds to the question and topics in hand, keeping the scope narrow while also giving the relevant answer.

ChatGPT managed to give more complete answers most of the time, although it should be stated that the length of and openness of the question very much impacted the results of the answers received from our implementation, regularly surpassing the ChatGPT pretrained results.

One important sighting in our testing was the simple fact that the more keywords, and the more specific they were - the better and more broad the answers were.

If there was something to be expected in the answer, including it in the keywords section would give a significant boost to the context and the broad significance of the answer.

Test query example

The testing query shown here is a simple example of the testing suite we have constructed. The query is a simple question about the use of diagrams and business processes in the context of the keywords given. The keywords are a simple set of words that are relevant to the question and the context of the answer. With these here we examplify the search of niche topic and papers not frequently mentioned in most computer science projects, showing the usefulness of agents and how our pipeline matches up with other agents.

Keep in mind that the question was posed to the other agents prefixed with ”Answer the question shortly” as well as without the keywords, keeping the focus on imporving basic search and retrieval of scientific information.

Keywords bpmn uml business processes

Question How and why are diagrams utilized in optimizing business processes?

ChatGPT Diagrams are used in optimizing business processes to visually map workflows, identify inefficiencies, and clarify roles, making it easier to analyze and improve operations for greater efficiency and effectiveness.

Claude Diagrams optimize business processes by: 1. Visualizing complex workflows to identify bottlenecks and inefficiencies 2. Enabling easier communication of processes across teams and stakeholders 3. Standardizing documentation using formats like flowcharts, BPMN, and value stream maps 4. Supporting process analysis and simulation before implementation 5. Facilitating continuous improvement through visual representation of ”before” and ”after” states. Diagrams transform abstract processes into concrete visuals, making problems more identifiable and solutions more implementable.

Baseline Diagrams are utilized in optimizing business processes to provide a visual representation of the process flow. This allows for easier analysis and identification of bottlenecks or inefficiencies in the process. Additionally, diagrams can help to standardize the process and ensure consistency across different stages of the process.

Future work

Although it is clear that our model got significant help, it shows promising results of our findings - searching and integrating proper papers into the agent pipeline gives significant importance to answers that academics would expect.

The next obvious step is to simply make the agent help himself, and automate the process of finding the keywords. Several testing suits should be made but simply extracting the keywords from a query, when none are given, is a clear first step in this implementation

Storing and thinking of better ways of analyzing these papers is also a proper view to take. Taking a step into the long term use and further improvements of the model, it could learn and store facts and papers it has deemed important for future sort of caching of knowledge and improving the vision it could have on the search.

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam blandit dictum facilisis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam convallis tellus velit, quis ornare ipsum aliquam id. Maecenas tempus mauris sit amet libero elementum eleifend. Nulla nunc orci, consectetur non consequat ac, consequat non nisl. Aenean vitae dui nec ex fringilla malesuada. Proin elit libero, faucibus eget neque quis, condimentum laoreet urna. Etiam at nunc quis felis pulvinar dignissim. Phasellus turpis turpis, vestibulum eget imperdiet in, molestie eget neque. Curabitur quis ante sed nunc varius dictum non quis nisl. Donec nec lobortis velit. Ut cursus, libero efficitur dictum imperdiet, odio mi fermentum dui, id vulputate metus velit sit amet risus.

Equations

You can write equations inline, e.g. $\cos\pi = -1$, $E = m \cdot c^2$ and α, or you can include them as separate objects. The Bayes’s rule is stated mathematically as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{1}$$

where A and B are some events. You can also reference it – the equation 1 describes the Bayes’s rule.

Lists

We can insert numbered and bullet lists:

1. First item in the list.
2. Second item in the list.
3. Third item in the list.

- First item in the list.
- Second item in the list.
- Third item in the list.

We can use the description environment to define or describe key terms and phrases.

Word What is a word?.

Concept What is a concept?

Idea What is an idea?

Random text

This text is inserted only to make this template look more like a proper report. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam blandit dictum facilisis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam convallis tellus velit, quis ornare ipsum aliquam id. Maecenas tempus mauris sit amet libero elementum eleifend. Nulla nunc orci, consectetur non consequat ac, consequat non nisl. Aenean vitae dui nec ex fringilla malesuada. Proin elit libero, faucibus eget neque quis, condimentum laoreet urna. Etiam at nunc quis felis pulvinar dignissim. Phasellus turpis turpis, vestibulum eget imperdiet in, molestie eget neque. Curabitur quis ante sed nunc varius dictum non quis nisl. Donec nec lobortis velit. Ut cursus, libero efficitur dictum imperdiet, odio mi fermentum dui, id vulputate metus velit sit amet risus. Nulla vel volutpat elit. Mauris ex erat, pulvinar ac accumsan sit amet, ultrices sit amet turpis.

Phasellus in ligula nunc. Vivamus sem lorem, malesuada sed pretium quis, varius convallis lectus. Quisque in risus nec lectus lobortis gravida non a sem. Quisque et vestibulum sem, vel mollis dolor. Nullam ante ex, scelerisque ac efficitur vel, rhoncus quis lectus. Pellentesque scelerisque efficitur purus in faucibus. Maecenas vestibulum vulputate nisl sed vestibulum. Nullam varius turpis in hendrerit posuere.

Figures

You can insert figures that span over the whole page, or over just a single column. The first one, Figure 1, is an example of a figure that spans only across one of the two columns in the report.

On the other hand, Figure 2 is an example of a figure that spans across the whole page (across both columns) of the report.

Tables

Use the table environment to insert tables.

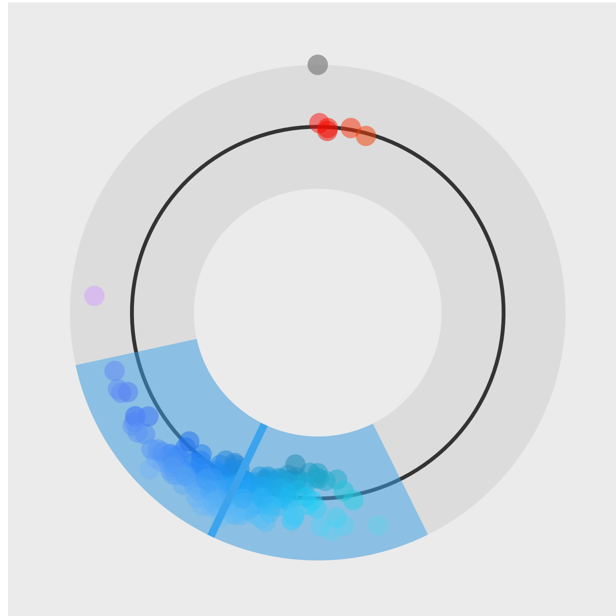

Figure 1. A random visualization. This is an example of a figure that spans only across one of the two columns.

Table 1. Table of grades.

Name		
First name	Last Name	Grade
John	Doe	7.5
Jane	Doe	10
Mike	Smith	8

Code examples

You can also insert short code examples. You can specify them manually, or insert a whole file with code. Please avoid inserting long code snippets, advisors will have access to your repositories and can take a look at your code there. If necessary, you can use this technique to insert code (or pseudo code) of short algorithms that are crucial for the understanding of the manuscript.

Listing 1. Insert code directly from a file.

```
import os
import time
import random

fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

Listing 2. Write the code you want to insert.

```
import(dplyr)
import(ggplot)

ggplot(diamonds,
          aes(x=carat, y=price, color=cut)) +
  geom_point() +
```

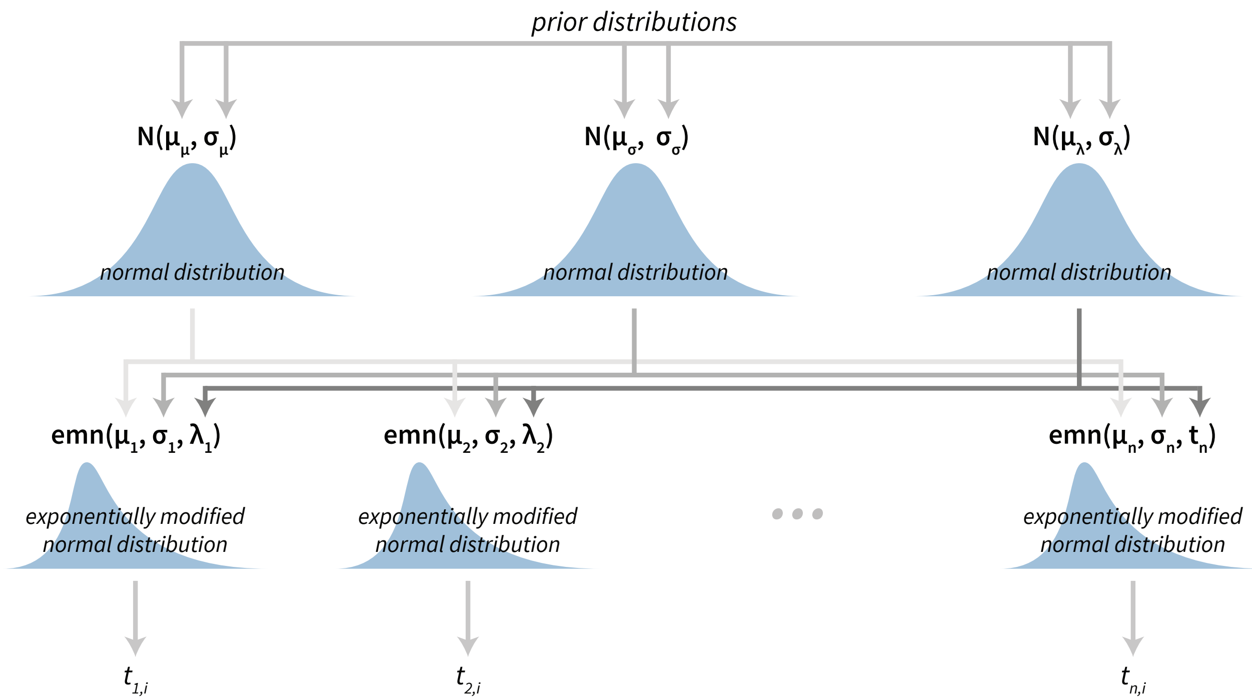

Figure 2. Visualization of a Bayesian hierarchical model. This is an example of a figure that spans the whole width of the report.

```
geom_smooth()
```

Results

Use the results section to present the final results of your work. Present the results in a objective and scientific fashion. Use visualisations to convey your results in a clear and efficient manner. When comparing results between various techniques use appropriate statistical methodology.

More random text

This text is inserted only to make this template look more like a proper report. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam blandit dictum facilisis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam convallis tellus velit, quis ornare ipsum aliquam id. Maecenas tempus mauris sit amet libero elementum eleifend. Nulla nunc orci, consectetur non consequat ac, consequat non nisl. Aenean vitae dui nec ex fringilla malesuada. Proin elit libero, faucibus eget neque quis, condimentum laoreet urna. Etiam at nunc quis felis pulvinar dignissim. Phasellus turpis turpis, vestibulum eget imperdiet in, molestie eget neque. Curabitur quis ante sed nunc varius dictum non quis nisl. Donec nec lobortis velit. Ut cursus, libero efficitur dictum imperdiet, odio mi fermentum dui, id vulputate metus velit sit amet risus. Nulla vel volutpat elit. Mauris ex erat, pulvinar ac accumsan sit amet, ultrices sit amet turpis.

Phasellus in ligula nunc. Vivamus sem lorem, malesuada sed pretium quis, varius convallis lectus. Quisque in risus nec lectus lobortis gravida non a sem. Quisque et vestibulum sem, vel mollis dolor. Nullam ante ex, scelerisque ac efficitur vel, rhoncus quis lectus. Pellentesque scelerisque efficitur purus in faucibus. Maecenas vestibulum vulputate nisl sed vestibulum. Nullam varius turpis in hendrerit posuere.

Nulla rhoncus tortor eget ipsum commodo lacinia sit amet eu urna. Cras maximus leo mauris, ac congue eros sollicitudin ac. Integer vel erat varius, scelerisque orci eu, tristique purus. Proin id leo quis ante pharetra suscipit et non magna. Morbi in volutpat erat. Vivamus sit amet libero eu lacus pulvinar pharetra sed at felis. Vivamus non nibh a orci viverra rhoncus sit amet ullamcorper sem. Ut nec tempor dui. Aliquam convallis vitae nisi ac volutpat. Nam accumsan, erat eget faucibus commodo, ligula dui cursus nisi, at laoreet odio augue id eros. Curabitur quis tellus eget nunc ornare auctor.

Discussion

Use the Discussion section to objectively evaluate your work, do not just put praise on everything you did, be critical and exposes flaws and weaknesses of your solution. You can also explain what you would do differently if you would be able to start again and what upgrades could be done on the project in the future.

Acknowledgments

Here you can thank other persons (advisors, colleagues ...) that contributed to the successful completion of your project.

References

[1] 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.

[2] Hangfeng He, Hongming Zhang, and Dan Roth. Rethinking with retrieval: Faithful large language model inference, 2022.

[3] Sangita Pokhrel, Bina K C, and Prashant Bikram Shah. A practical application of retrieval-augmented generation for website-based chatbots: combining web scraping, vectorization, and semantic search. *Journal of Trends in Computer Science and Smart Technology*, 6(4):424–442, January 2025.

[4] Nikunj Kanataria, Kunj Pareshbhai Patel, Hetul Niteshbhai Patel, Parth Goel, Krishna Patel, and Dweepna Garg. Rag-enhanced large language model for intelligent assistance from web-scraped data. In *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, pages 1043–1048, 2024.