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

# Conversational Agent with Retrieval-Augmented Generation

Luka Dragar, Urša Keše, Ožbej Pavc

**Abstract**

This project aims to develop an advanced conversational agent that leverages Retrieval-Augmented Generation (RAG) to enhance response quality and factual accuracy. Unlike traditional chatbots relying solely on pre-trained knowledge, our system will incorporate real-time web scraping capabilities to retrieve up-to-date information from the internet. By dynamically fetching and integrating external information during conversation, the agent can provide more accurate, comprehensive, and current responses. This preliminary report outlines our proposed methodology, including the design of a multi-component pipeline encompassing web scraping, content processing, and LLM integration. We discuss relevant datasets for evaluation and present our planned approach for assessing the system's performance using both automated metrics and human evaluation.

**Keywords**

Retrieval-Augmented Generation, Conversational AI, Real-time Information Retrieval, Web Scraping, Natural Language Processing, Knowledge-Augmented Chatbots

*Advisors: Aleš Žagar*

## Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities in generating human-like text and engaging in conversations. However, they face significant limitations, particularly in providing up-to-date information and ensuring factual accuracy. These models are trained on historical data and lack access to real-time information, leading to potential outdated responses or "hallucinations" where they confidently generate incorrect information.

Retrieval-Augmented Generation (RAG) has emerged as a promising approach to address these limitations by augmenting LLMs with external knowledge sources. Traditional RAG systems typically rely on pre-indexed document collections or knowledge bases. Our project extends this concept by incorporating real-time web scraping capabilities, allowing the conversational agent to retrieve and integrate the most current information available online.

The project has several key objectives:

1. Design and implement a comprehensive pipeline for web scraping and content processing
2. Develop effective techniques for integrating retrieved information into LLM prompts
3. Create mechanisms for tracking and citing sources to ensure transparency
4. Evaluate the system's performance in terms of response quality, factual accuracy, and relevance

This approach has wide-ranging applications, including customer support, academic research assistance, news summarization, and general knowledge queries where freshness and accuracy of information are critical. By enhancing conversational agents with the ability to retrieve and incorporate external knowledge in real-time, we aim to bridge the gap between static pre-trained knowledge and dynamic information needs.

## Related Work

#### 0.0.1 Retrieval-Augmented Generation for AI-Generated Content: A Survey

The paper provides a comprehensive review of Retrieval-Augmented Generation (RAG), a paradigm that enhances AI-generated content by integrating information retrieval into the generation process. The authors classify RAG into four foundational paradigms based on how retrieval augments generation: query-based, latent representation-based, logit-based, and speculative RAG.

They also discuss various enhancements to improve RAG systems, such as input optimization, retriever and generator fine-tuning, and pipeline improvements. The survey covers RAG applications across multiple modalities (text, code,

knowledge, images, videos, audio, 3D, and science) and highlights benchmarks for evaluating RAG systems.

Finally, the paper identifies current limitations (e.g., noise in retrieval, system complexity) and suggests future research directions, including novel augmentation methodologies, flexible pipelines, and broader applications. The goal is to provide a unified perspective on RAG, facilitating its development and deployment in AI-generated content tasks. [1]

#### 0.0.2 Benchmarking Large Language Models in Retrieval-Augmented Generation

Chen et al. propose a comprehensive benchmark for evaluating LLMs in RAG scenarios. Their work highlights the challenges in assessing how effectively models utilize retrieved information and identifies key performance factors. The authors demonstrate that while larger models generally perform better in RAG tasks, the quality of retrieved information significantly impacts overall performance. Their benchmark provides valuable metrics for comparing different RAG implementations and offers insights into optimization strategies. This research informs our approach to evaluation and helps establish meaningful baselines for assessing our web scraping-enhanced RAG system. [2]

#### 0.0.3 FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation

The paper addresses a critical limitation of LLMs: their inability to access up-to-date information. The authors introduce FreshQA, a benchmark containing 600 diverse questions that test models' ability to handle current world knowledge and identify false premises. Through extensive evaluation of various LLMs, they demonstrate that all models struggle with questions requiring recent information and those with false premises. To address these limitations, researchers developed FreshPrompt, a method that enhances LLM performance by incorporating relevant search engine results into prompts. This approach significantly improves accuracy across all question types, with GPT-4+FreshPrompt achieving 77.6% accuracy under strict evaluation criteria. The authors' analysis reveals that the number and ordering of retrieved evidence pieces significantly impact performance, while concise answers help reduce hallucination compared to verbose ones. The paper contributes to the growing field of retrieval-augmented generation by providing both a benchmark for evaluating LLMs' factuality and a simple yet effective method for enhancing their performance with external knowledge. [3]

#### 0.0.4 A Multi-Source Retrieval Question Answering Framework Based on RAG

The authors present a novel multi-source retrieval framework called MSRAG (Multi-Source Retrieval-Augmented Generation) to enhance the accuracy and relevance of question-answering (QA) systems. Traditional Retrieval-Augmented Generation (RAG) methods often suffer from irrelevant or noisy retrieval information, leading to unreliable answers. To address this, the authors propose integrating GPT-3.5 for retrieval and semantic segmentation of queries, alongside web retrieval, to improve the granularity and relevance of retrieved information. The framework combines GPT retrieval, web retrieval, and non-retrieval strategies, using a loss function to select the most accurate answer from multiple sources. Experiments on knowledge-intensive QA datasets (2WikiMultiHopQA, HotpotQA, and StrategyQA) demonstrate that MSRAG outperforms existing RAG frameworks in terms of efficiency and accuracy. The study highlights the potential of using GPT-3.5 as a substitute for traditional retrievers and emphasizes the benefits of integrating multiple retrieval sources to mitigate noise and improve retrieval relevance. Future research directions include optimizing GPT retrieval performance and reducing operational costs in integrated approaches. [4]

#### 0.0.5 When Search Engine Services meet Large Language Models: Visions and Challenges

The paper discusses the relationship between search engines and Large Language Models (LLMs). It examines two main research directions: 1. Search4LLM: How search engines can enhance LLMs by providing diverse datasets for pre-training, supplying relevant documents to improve query answering, using Learning-To-Rank tasks to enhance precision, and incorporating recent search results for more accurate content. 2. LLM4Search: How LLMs can improve search engines through better content summarization for indexing, query optimization, enhanced ranking through document relevance analysis, and data annotation for learning-to-rank tasks. The authors identify key challenges including addressing biases in training models, managing computational costs, and continuously updating LLM training with changing web content. They also discuss broader implications for service computing such as scalability, privacy concerns, and the need to adapt search engine architectures for these advanced models. The paper provides a comprehensive framework for understanding how these technologies can complement each other, potentially transforming how users interact with information online while highlighting the technical and ethical considerations that must be addressed. [5]

## Preliminary Ideas and Approach

### System Architecture

Our proposed system follows a pipeline architecture with several interconnected components:

1. **Query Processing Module**: Analyzes user input to identify information needs and formulates appropriate search queries. For complex questions, this module may decompose the query into multiple sub-queries.
2. **Web Search Interface**: Connects with search engines to identify relevant web pages based on the processed queries. This component optimizes search terms to improve result quality.
3. **Web Scraping Engine**: Extracts content from identified web pages, handling different HTML structures. This includes rate limiting and proper error handling.

4. **Content Processing**: Cleans and normalizes the scraped content, removing irrelevant elements like advertisements and navigation menus. This component also performs summarization and extracts key information relevant to the user query.
5. **Chunking & Embedding System**: Divides processed content into semantic chunks and generates vector embeddings for efficient retrieval. These embeddings are stored and ranked based on relevance to the query.
6. **LLM Integration Layer**: Constructs prompts incorporating the most relevant retrieved content and manages the interaction with the LLM. This component also tracks citation information to ensure proper attribution.
7. **Orchestration Service**: Coordinates the entire pipeline flow, managing errors and implementing caching strategies for performance optimization.

### Datasets

For training and evaluation, we plan to utilize the following datasets:

- **MS MARCO**: A large-scale dataset containing real search queries and human-generated answers, useful for training query understanding and answer generation components.
- **Natural Questions (NQ)**: Contains real queries from Google Search and corresponding Wikipedia articles with annotated answers, providing a good baseline for question-answering performance.
- **TempLAMA**: A dataset specifically designed for evaluating temporal knowledge in language models, which will help assess our system's ability to provide up-to-date information.
- **Custom Benchmark**: We plan to create a custom benchmark of time-sensitive queries requiring recent information, which will specifically test our web scraping-enhanced RAG capabilities against traditional approaches.

### Implementation Considerations

Our preliminary implementation plans include:

- Using BeautifulSoup and Selenium for web scraping to handle both static and dynamic content
- Implementing a vector database (e.g., ChromaDB or FAISS) for efficient storage and retrieval of embeddings
- Developing prompt templates optimized for incorporating external information
- Creating a robust evaluation framework that measures both quantitative metrics and qualitative aspects of responses

### Evaluation Strategy

We will evaluate our system using multiple approaches:

1. **Component-level Testing**: Assessing each pipeline component independently (e.g., scraper accuracy, chunking quality)
2. **Automated Metrics**: Using established metrics such as ROUGE, BLEU, and BERTScore to evaluate response quality
3. **Factual Accuracy**: Developing specialized metrics to evaluate the correctness of information provided
4. **Comparative Analysis**: Comparing our web scraping-enhanced RAG system against other commercial LLMs with web search ability.
5. **Human Evaluation**: Conducting user studies to assess subjective aspects like helpfulness, coherence, and overall satisfaction

### Challenges and Considerations

We anticipate several challenges:

- Managing website access limitations and respecting terms of service
- Efficiently handling different website structures and content formats
- Balancing the size of retrieved content with the context limitations of LLMs
- Ensuring proper attribution and citation of sources
- Optimizing performance for real-time conversation

## References

[1] Yunfan Zhao, Junjie Gao, Yijian Wu, Zihan Weng, Zeyuan Zhang, Xiaoming Li, Huaishao Lin, Xianpei Han, and Le Sun. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint*, arXiv:2312.10997, 2024.

[2] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking Large Language Models in Retrieval-Augmented Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17754–17762, 2024.

[3] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. Freshllms: Refreshing large language models with search engine augmentation, 2023.

[4] Ridong Wu, Shuhong Chen, Xiangbiao Su, Yuankai Zhu, Yifei Liao, and Jianming Wu. A multi-source retrieval question answering framework based on rag, 2024.

[5] Haoyi Xiong, Jiang Bian, Yuchen Li, Xuhong Li, Mengnan Du, Shuaiqiang Wang, Dawei Yin, and Sumi Helal. When search engine services meet large language models: Visions and challenges, 2024.