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 We demonstrate how this approach addresses common limitations of traditional language models, such as outdated information and hallucinations, while providing targeted academic research assistance.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

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 analize 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.

Methodology

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.

Vectorization and FAISS

The vectorization of the data is done using the FAISS library, which allows us to efficiently search and retrieve similar papers based on the keywords given in the query.

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 . The k parameter is the amount of papers that are being fetched from the vectorized data and used to answer the question. This is something that could be further analyzed and tested, as it is a very important part of the pipeline.

Conversational agent and the pipeline

The conversational agent is implemented using the LangChain library, which allows us to easily create a pipeline that can handle the queries and return the answers based on the vectorized data.

Methodology - Anja

Baseline Model Implementation

In the sample work of our proof-of-concept model setup, the focus was mainly on the various resources and ways of implementing them.

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.

Database

Creation: papers, resources and fetching

We've managed to group together and take use of many knowledge bases of academic research and compile them into the sample RAG pipeline that manages to access these papers in a keyword fashion and give answers enriched with these resources.

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 implemented in the test of the capabilities of the pipeline.

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

Popular and massive academic libraries such as Arxiv and Semantic Scholar already have available Python libraries allowing us to 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.

Dataset Details

how many ppaers, from how many keywords etc

Expanding the Baseline with Database

The vectorization of the data is done using the FAISS library, which allows us to efficiently search and retrieve similar papers based on the keywords given in the query.

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.

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 implemented to our answers.

The k parameter is the amount of papers that are being fetched from the vectorized data and used to answer the question.

This is something that could be further analyzed and tested, as it is a very important part of the pipeline.

On-Demand Paper Fetching

The user can choose the option to fetch additional papers on demand....

Prompt Design

...

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.

Evaluation

Since the goal of this project is to assist students and academic staff with their research, we believe it's important to include their feedback in the evaluation process. That's why our evaluation is mainly experimental and focused on how real users interact with and judge the system.

The evaluation of our model was mostly done through experiments that tested how well it performs in different settings. We focused on how helpful, clear, and relevant the model's answers were, and we compared it to other AI models

Model comparison

This section presents a direct comparison between our model and other well-known conversational agents to better understand its strengths and weaknesses in academic research contexts.

A selection of responses from all four models (our two modes, ChatGPT, and Claude) for a subset of prompts is included in Appendix A.

Evaluation Through User Feedback

To better understand how our model performs in realistic academic settings, we created a questionnaire aimed at students with basic knowledge of Natural Language Processing (NLP). The goal was to gather feedback from actual users—students who are likely to use such a tool in their own research. We asked them to rank the quality of answers generated by our model in two different modes, and compare those with a third response generated by ChatGPT, a widely used conversational agent.

We designed the questionnaire to compare three types of responses to the same prompts:

- *Our baseline model*, which uses only the data stored in our vectorstore and does not search for new papers.

- *Our extended model*, which includes live retrieval of new academic papers based on the keywords in the user's query.

- *ChatGPT*, used as a comparison to see how our model performs against a popular conversational agent.

The questions or prompts in the questionnaire were designed to simulate real academic research scenarios, similar to how a student might ask for help when researching a topic. This made the evaluation more realistic and aligned with how the system is actually intended to be used. Since ChatGPT often provides longer answers, we asked it to match the expected length of our model's responses to keep the comparison fair.

Participants were then asked to rank the three responses (1 being best, 3 being worst) based on relevance, clarity, and usefulness. This allowed us to collect both quantitative and qualitative feedback on how well each model performed.

Overall, this evaluation approach gave us valuable insights into the strengths and weaknesses of our model in realistic academic use cases. It also showed the added value of including up-to-date papers, especially in fields where recent research is important.

Comparison with ChatGPT and Claude

To complement the user-based evaluation, we included a qualitative comparison between our model and two widely used conversational agents: ChatGPT and Claude. For each prompt used in the questionnaire, we generated responses from all three systems: our baseline model, our extended retrieval-augmented model, and the external agents. These outputs were then examined side-by-side to assess differences in style, structure, and content delivery. While this comparison was not part of the formal user evaluation, it provided an additional perspective on how our models align with or differ from established systems in terms of generation technique and retrieval integration.

Comparing and discussing different possible system prompts

In this section, we discuss the impact of different system prompts on the performance of our conversational agent.

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.

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.

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.

Conclusion

The conclusion.

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.

Appendix A: Answer Comparisons

Question 1: What is natural language processing

- **Without New Papers:** Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language. It involves the application of computational techniques to the analysis, synthesis, and comprehension of human language. NLP is used in various fields such as speech recognition, sentiment analysis, machine translation, and text summarization.

- **With New Papers:** Natural language processing (NLP) is a subfield of artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language. It involves the use of computational techniques to analyze and represent naturally occurring texts, at one or more levels of linguistic analysis, for the purpose of achieving human-like language processing for a range of tasks or applications. NLP algorithms are not naturally designed to process language, so they require significant amounts of training data and sophisticated computational methods to achieve accurate results.

- **ChatGPT Answer:** Natural Language Processing (NLP) is a branch of artificial intelligence that focuses on the interaction between computers and human language. It enables machines to understand, interpret, and respond to text or speech. Common applications include chatbots, translation, and sentiment analysis.

- **Claude Answer:** Natural language processing (NLP) is a branch of artificial intelligence that focuses on the interaction between computers and human language. It involves developing algorithms and technologies that enable computers to understand, interpret, and generate human language in a way that is both meaningful and useful.

Question 2: What are some recent papers published in the Natural Language Processing domain

- **Without New Papers:** Sure! Here are some recent papers published in the Natural Language Processing domain:
  1. *Multi-step Moral Dilemmas* by [Author Name] –

This paper introduces a new dataset specifically constructed to evaluate the evolving moral judgments of LLMs across 3,302 five-stage dilemmas. The study reveals that LLMs recalibrate their moral judgments based on scenario complexity and highlights the dynamic and context-dependent nature of LLM ethical reasoning.

2. *Deep Generative Models* by [Author Name] – This survey presents a brief description of the advances that have occurred in the area of deep generative modeling. It covers topics such as language modeling, text generation, and image synthesis.

3. *Natural Language Understanding* by [Author Name] – This paper focuses on building language understanding models, which involve tasks such as named entity recognition, sentiment analysis, and question answering. The study explores different approaches to language understanding and their performance on various benchmark datasets.

- **With New Papers:** Here are some recent papers published in the Natural Language Processing domain:
  - **Title:** Multi-step Moral Dilemmas for Evaluating LLM Ethical Reasoning
  Author: Xiaoyan Zhang, et al.
  Overview: Introduces the MMDs dataset to evaluate evolving moral judgments of LLMs across 3,302 five-stage dilemmas.
  - **Title:** Mining Unstructured Text Data for Smart Construction Project Management
  Author: Yongjin Kim, et al.
  Overview: Proposes a novel NLP method to mine unstructured text for smart project management in construction.
  - **Title:** Improving Named Entity Recognition with Deep Learning
  Author: Jianxin Chen, et al.
  Overview: Suggests a deep learning approach for enhancing NER accuracy, outperforming state-of-the-art models.

- **ChatGPT Answer:** Sure! Here are some recent papers published in the Natural Language Processing domain:
  1. *LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders* – Proposes LLM2Vec, a method for turning decoder-only LLMs into strong encoders.
  2. *Med-Gemini: Capabilities of Gemini Models in Medicine* – Introduces Med-Gemini, a multimodal model for healthcare, outperforming GPT-4.
  3. *MASIVE: Open-Ended Affective State Identification in English and Spanish* – Presents a dataset and method for nuanced emotion detection.
  4. *GeoGPT4V: Towards Geometric Multi-modal Large Language Models* – Focuses on enhancing LLMs in solving geometry problems with visual data.
  5. *Enhancing Machine Translation for Low-resource*

Languages: A Cross-lingual Learning Approach for TWI – Offers a framework for improving translation in low-resource languages.

- **Clause Answer:** Based on my search, here are some notable recent papers and developments in Natural Language Processing:
  "Large Language Models: New Opportunities for Access to Science" (January 2025) - This paper explores how LLMs like ChatGPT are being adapted for scientific information retrieval and making science more accessible.
  "Large language models for automated scholarly paper review: A survey" (January 2025) - Examines the growing use of LLMs in academic peer review processes and automated scholarly paper review systems.
  "Large Language Models Meet NLP: A Survey" - Provides a comprehensive overview of how LLMs are currently applied to NLP tasks and explores whether traditional NLP problems have been solved.
  "Recent Advances in Generative AI and Large Language Models: Current Status, Challenges, and Perspectives" - Explores the current state of generative AI and LLMs, their applications, and emerging challenges.

  Major upcoming conferences include RANLP 2025 (September 8-10), EMNLP 2025 (November 5-9 in Suzhou, China), and ICNLP 2025. Current research focuses on Large Language Model applications and integration with traditional NLP tasks, low-resource language processing and multilingual models, automated scientific paper review and academic applications, Retrieval Augmented Generation for scientific information access, and efficiency and optimization of transformer architectures.

Question 3: How would I use ML to reconstruct damaged artwork?

- **Without New Papers:** Machine learning (ML) can be used to reconstruct damaged artwork by analyzing the available information and generating a close approximation of the original. Techniques include:
  1. **Image restoration:** Using trained models (like CNNs) to predict and fill in missing pixels based on surrounding areas.
  2. **Image synthesis:** Using Generative Adversarial Networks (GANs) to generate new images by learning from undamaged examples.
  3. **Transfer learning:** Fine-tuning pre-trained models on a dataset of damaged artwork to improve reconstruction performance.

- **With New Papers:** Several methods are used for artwork restoration using ML. CNNs can be trained on undamaged images to recreate damaged parts, while GANs are used to generate new images that match the

style and structure of the original. Some algorithms are specifically tailored for image inpainting and restoration.

- **ChatGPT Answer:** To reconstruct damaged artwork with ML, image inpainting techniques using CNNs or GANs are applied. These models are trained on undamaged artwork to learn patterns and generate realistic restorations. Style transfer can also ensure visual consistency with the original.

- **Claude Answer:** Machine learning can be applied to artwork reconstruction through several sophisticated approaches:
  Computer Vision Techniques: Convolutional Neural Networks (CNNs) can be trained to identify patterns, textures, and artistic styles in undamaged portions of artwork, then predict what missing or damaged areas should look like. Image inpainting algorithms use encoder-decoder architectures to fill gaps based on surrounding context.
  Generative Adversarial Networks (GANs): These can generate realistic artistic content by learning the statistical patterns of an artist's style from their complete works. The generator creates plausible reconstructions while the discriminator ensures they match the original artistic style and period.
  Style Transfer Networks: These networks can separate content from style, allowing reconstruction that maintains the original composition while applying the correct artistic technique and color palette.
  Multi-modal Learning: Combining visual analysis with historical records, X-ray imaging, and infrared photography provides additional context for more accurate reconstruction.

Question 4: I want to start learning about RAG in Natural Language Processing. What literature and articles do you suggest I go through? Can you give me dates of their release as well?

- **Without New Papers:**
  There are several papers and articles that discuss RAG (Role-Aware Generation) in Natural Language Processing. Here are some suggestions:

  – "Role-Aware Generation" by Yoshua Bengio, et al., published in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). This paper introduces the concept of role-aware generation and presents a model that generates text based on the roles of the entities involved.
  – "Role-Aware Text Generation with Attention Mechanisms" by Zhongyuan Liu, et al., published in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP). This paper presents a model that uses attention mechanisms to generate text based on the roles of the entities involved.

– "Role-Aware Text Generation using Transformers" by Xiaoyu Zhang, et al., published in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP). This paper presents a model that uses transformers to generate text based on the roles of the entities involved.

These papers were released between 2015 and 2019.

- **With New Papers:**
  Here are five recent literature and articles related to RAG in Natural Language Processing that you can consider reading:

  1. "Prompting-Based Methods for Reasoning with Language Models" by Jiang et al. (2023). This paper proposes a prompting-based method to enable reasoning with language models.
  2. "A Survey of Reasoning with Language Models" by Zhang et al. (2023). This literature review focuses on the state-of-the-art methods for reasoning with language models.
  3. "Noise on Reasoning with Generative Adversarial Networks" by Chen et al. (2023). This paper investigates the impact of noise on reasoning with generative adversarial networks.
  4. "Fine-Tuning Language Models with New Factual Data" by Li et al. (2023). This paper explores the effects of fine-tuning language models with new factual data.
  5. "Reasoning with Language Models: A Comprehensive Review" by Wang et al. (2023). This literature review provides a comprehensive overview of reasoning with language models.

  All of these papers were released in 2023 or later.

- **ChatGPT Answer:**
  Unsupervised learning in NLP involves training models on unlabeled text to discover patterns without explicit annotations. Common techniques include word embeddings (like Word2Vec), topic modeling (e.g., LDA), clustering, and pretraining language models (like BERT and GPT) using tasks such as masked word prediction. These methods help with tasks like text summarization, sentiment analysis, and improving search by learning from raw text data.

  For more info, check these links:
  – GeeksforGeeks on Unsupervised Learning
  – Wikipedia Unsupervised Learning
  – ArXiv paper on Unsupervised Text Classification
  Let me know if you want details on any specific method!

- **Claude Answer:**

  Foundational Papers:

  – *"Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"* (May 2020) – The original

RAG paper that introduced the concept of combining pre-trained parametric and non-parametric memory. [2005.11401]

Comprehensive Surveys:

– *"Retrieval-Augmented Generation for Large Language Models: A Survey"* (December 2023) – Covers Naive RAG, Advanced RAG, and Modular RAG paradigms. [2312.10997]

– *"A Comprehensive Survey of RAG: Evolution, Current Landscape and Future Directions"* (October 2024) – Traces RAG evolution from foundational concepts to current state of the art. (PDF)

Recent Advanced Methods:

– *"CRAG: Corrective Retrieval Augmented Generation"* (2024) – Improves robustness when retrieval fails by assessing document quality. *Retrieval Augmented Generation Research: 2017–2024*

– *"SELF-RAG: Self-Reflective Retrieval-Augmented Generation"* (2024) – Incorporates self-reflective mechanism that dynamically decides when to retrieve. *The 2025 Guide to Retrieval-Augmented Generation (RAG)*

– *"MiniRAG"* (January 2025) – Lightweight RAG system for small language models with semantic-aware graph indexing. *Top 10 RAG Papers from January 2025*

Question 5: What can you tell me about semantic web? Give me longer, more informed answer.

- **Without New Papers:**
The Semantic Web is a vision for the future of the World Wide Web, in which data is represented using machine-readable formats and processed by machines rather than humans. It is built on top of the existing Web infrastructure and aims to enable computers to understand and interpret information on the Web in a way that is currently not possible.

The core technology behind the Semantic Web is the Resource Description Framework (RDF), which provides a standard way of representing data as triples (subject-predicate-object statements). RDF allows data to be linked together in a graph-like structure, making it easier for machines to reason about relationships between different pieces of information.

Another key component of the Semantic Web is the use of ontologies, which provide a formal definition of the meaning of terms and relationships within a specific domain. Ontologies allow machines to understand the meaning of words and concepts in a consistent and precise way, enabling them to reason about information in a more sophisticated manner.

There are several ongoing efforts to develop and deploy the Semantic Web, including the creation of standardized vocabularies and ontologies, the development of tools and platforms for working with RDF and ontologies, and the integration of the Semantic Web with existing Web technologies such as HTTP and HTML.

While the Semantic Web is still in its early stages, it has the potential to revolutionize the way we interact with information on the Web, enabling machines to understand and interpret information in a more intelligent and intuitive way. This could have a wide range of applications, from improving search and information retrieval to enabling new forms of collaboration and knowledge sharing.

- **With New Papers:**
The Semantic Web is a vision for the future of the World Wide Web, in which data is represented using machine-readable formats and processed by machines rather than humans. It is built on top of the existing Web infrastructure and aims to enable computers to understand and interpret information on the Web in a way that is currently not possible.

The core technology behind the Semantic Web is the Resource Description Framework (RDF), which provides a standard way of representing data as triples (subject-predicate-object statements). RDF allows data to be linked together in a graph-like structure, making it easier for machines to reason about relationships between different pieces of information.

Another key component of the Semantic Web is the use of ontologies, which provide a formal definition of the meaning of terms and relationships within a specific domain. Ontologies allow machines to understand the meaning of words and concepts in a consistent and precise way, enabling them to reason about information in a more sophisticated manner.

There are several ongoing efforts to develop and deploy the Semantic Web, including the creation of standardized vocabularies and ontologies, the development of tools and platforms for working with RDF and ontologies, and the integration of the Semantic Web with existing Web technologies such as HTTP and HTML.

While the Semantic Web is still in its early stages, it has the potential to revolutionize the way we interact with information on the Web, enabling machines to understand and interpret information in a more intelligent and intuitive way. This could have a wide range of applications, from improving search and information retrieval to enabling new forms of collaboration and knowledge sharing.

- **ChatGPT Answer:**

The Semantic Web is an evolution of the current World Wide Web that aims to make data on the internet understandable not just by humans but also by machines. While the traditional web focuses on linking documents primarily for human consumption, the Semantic Web adds a layer of meaning by structuring data with metadata and using shared vocabularies. This allows computers to interpret the context and relationships between data points, enabling more intelligent data processing, integration, and automation.

At the core of the Semantic Web are technologies like the Resource Description Framework (RDF), which represents information as triples—subject, predicate, and object—forming a graph of linked data. Ontologies provide formal definitions of concepts and relationships within specific domains, giving machines a shared vocabulary to understand the data consistently. To query this structured data, the Semantic Web uses SPARQL, a powerful language designed for retrieving and manipulating information stored in RDF format.

The benefits of the Semantic Web are wide-ranging. It enables improved search engines that understand user queries semantically rather than relying solely on keyword matching, resulting in more relevant and precise results. It also facilitates data integration from diverse sources, such as healthcare, finance, and social media, by providing a common framework to describe and connect data. Additionally, the Semantic Web supports automation through intelligent agents that can reason over linked data, enabling applications like virtual assistants or recommendation systems. Major tech companies leverage these principles to build knowledge graphs that enhance user experiences by organizing complex information about entities and their relationships.

Despite its potential, the Semantic Web faces challenges including the complexity of creating and maintaining ontologies, ensuring high-quality and trustworthy data, and achieving broad adoption among data providers. Nonetheless, ongoing developments and growing interest continue to push the Semantic Web closer to realizing its vision of a web of data that is as accessible and understandable to machines as it is to humans.

If you want to explore further, useful resources include the W3C Semantic Web Overview and Tim Berners-Lee's Linked Data design principles. Let me know if you'd like me to explain any part in more detail!

- **Claude Answer:**

The Semantic Web represents a fundamental evolution of the World Wide Web, transforming it from a collection of human-readable documents into a vast, machine-interpretable knowledge graph. Conceived by Tim Berners-Lee in the early 2000s, the Semantic Web aims to create a web of data that computers can understand, process, and reason about automatically.

Core Philosophy and Vision

The Semantic Web addresses a critical limitation of the traditional web: while humans can easily understand the meaning and context of web content, machines can only process the syntax and structure. The vision is to add a layer of meaning (semantics) to web data, enabling machines to understand not just what information is present, but what it means and how it relates to other information. This would allow for more intelligent search, automated reasoning, and seamless data integration across different systems and domains.

Foundational Technologies

The Semantic Web is built upon a stack of standardized technologies, each serving a specific purpose. At the base level, URIs (Uniform Resource Identifiers) provide unique identifiers for resources. XML provides the syntactic foundation, while RDF (Resource Description Framework) forms the core data model, representing information as subject-predicate-object triples. RDFS (RDF Schema) and OWL (Web Ontology Language) provide vocabulary and ontology capabilities, allowing for the definition of classes, properties, and relationships. SPARQL serves as the query language for RDF data, analogous to SQL for relational databases.

RDF and Knowledge Representation

RDF fundamentally changes how we think about data representation. Instead of storing information in tables or documents, RDF represents knowledge as a graph of interconnected statements. Each statement is a triple consisting of a subject (what we're talking about), a predicate (the property or relationship), and an object (the value or related entity). For example, "Shakespeare wrote Hamlet" becomes three components: Shakespeare (subject), wrote (predicate), and Hamlet (object). This model is incredibly flexible and allows for the representation of complex, interconnected knowledge in a way that's both human-understandable and machine-processable.