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 academic paper sources, dynamic web scraping capabilities and a pre-fetched knowledge base to build comprehensive answers to deep questions. The agent uses vector similarity search for context on topics and can be configured to search for additional papers online. We demonstrate how this approach addresses common limitations of traditional large language models, such as outdated information and uninformed responses, while providing targeted academic research assistance. We also highlight possibilities on future improvement on the concept and impact it could have on future research.

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.

We present a proof-of-concept model that demonstrates the potential of this approach, focusing on academic research. We will discuss the methodology, implementation details, and preliminary results of our model, which integrates RAG techniques with a knowledge base to provide far more informed answers.

Further, this paper highlights the possible limitations of the current implementation and it's pipeline, suggesting future improvements and directions, while also presenting a simple query and results, proving the impact such an agent could have further down the line, possibly more finely implemented and fine-tuned.

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

Pipeline overview

Our proposed method starts with building a database by collecting suitable articles from different sources. These documents are then embedded into vectors and queried based on their similarity score, using FAISS [5] and sentence-transformers. Document hashes are saved to a different file, to avoid adding duplicate entries. With this, we retrieve the most relevant documents using for our model. The selected articles are added as context to a pre-designed system template, which is then fed to a large language model to get the final answer. The details of our approach are explained in the following sections.

Baseline Model Implementation

During the work for the baseline implementation of our proof-of-concept model, our focus was mainly on the various resources and ways of implementing them. 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 can be obtained and have public API interfaces that give much leeway in how and what we search for. 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 Google Scholar 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. In our approach, we used it to web scrape papers from ResearchGate.

Document Storing and Retrieval

When storing the articles, we first have to obtain some metadata, such as the title, authors, year of publication, abstract and url. The latter could be used to access the page and read more about the article. During this, we also filter out the articles with empty abstracts, as they did not offer enough information for us to evaluate them fairly. Since most of the models are trained on data before 2023, we decided to only keep the documents that were published in year 2023 or later.

This being the first version of the model, the database was not yet set up, and documents were always fetched on the fly from all incorporated sources. In further iterations, we implemented the code that already contained document vectorization and FAISS library, as well as the baseline conversational agent with corresponding system prompt.

Issues and Limitations

While migrating the code to the Arnes HPC, there happened to be limitations. The Playwright library had problems with permissions in terms of browser access, which is why we were not able to use it's sources. We also concluded that this specific source was not as useful as we first thought, since it does not offer data about abstract, which is our most informative metadata. There were also issues with using Semantic Scholar, as the Python library resulted in calling SSL_CERTIFICATE, which we did not have access to. To obtain the relevant articles from Semantic Scholar, we had to run the code on our personal computers and transfer it to HPC via `scp` command.

We tested our baseline implementation by asking the model various questions. They did not have to be related to natural language processing, as the database was created temporarily for each run. The model took the question we posed as input and returned the answer, using the provided context from the fetched articles. The preliminary results were only made by subjective evaluation, and at the first glance, the model was returning meaningful and suitable results.

Database

In this subsection, we will highlight two different approaches we took to build the database of articles, which were used as the RAG context to the agent. The first approach was based on keywords, while the second one focused on filtering by categories. We will also describe how we created the documents and loaded them into the database.

Creation: papers, resources and fetching

Building with keywords. The first appraoch was to put together a database by selecting 241 keywords related to the field of natural language processing, which were assembled beforehand. We queried the model with: chunks of keywords (grouped by eg. 10 keywords with boolean OR into one query) and later, one by one. The documents were fetched from the same sources as described earlier. With each query, we attempted to find a maximum of 100 articles per source, per keyword, in order to enrich the database as much as possible. After some time, we document having issues with the two of the API endpoints, as they did denied the large number of queries that we were requesting.

The database ended up consisting of around 17000 articles. Within first tests, the similarity when retrieving documents turned out to be poor and in addition were completely unrelated to the question asked. One of the possible reasons was that our database ended up being too noisy, as the method with keywords did not capture suitable articles for the goal academic field. We decided to take a different approach, which is described in the following text.

Building with category filtering. Many of our previous sources did not have the ability to filter the articles by categories within the query, which is why we added some anew. Our main source this time was ACL, which contains papers from conferences in fields of natural language processing and linguistics. This provided us with around 15000 starting articles and the results improved immediately. Filtering by category was not necessary for ACL source, since the main field of study was already our main goal. To satisfy the need, we also modified an existing source (arXiv) so that the appropriate category was embedded in the query. OpenAlex has also been added as a source, as it is able to filter the articles by using concepts IDs. None of the listed retrievals were limited to a number of articles. They are, however, still to take into account the abstract and suitable publication year into account. This method got us a total of around 20000 articles.

Retriever

The retriever is the part of the pipeline that is responsible for fetching the relevant papers from the database based on the question posed. It uses the FAISS library to perform a similarity search on the vectorized data and returns the most relevant papers.

Although the retriever is a simple part of the pipeline, it is very important for the overall performance of the model. The retriever is responsible for fetching the relevant papers from the database and providing them to the model as context for answering the question.

Our implementation uses the FAISS library to perform a similarity search on the vectorized data and returns the most relevant papers based on the question posed. The retriever is also responsible for scoring the papers based on their relevance to the question, which is in this case done on simple similarity search. We've configured it as such so that it returns the top 4 most relevant papers it could find in the database. We've found this was a good number to use, as it provided enough context for the model to generate a relevant answer, while also not being too many papers to overwhelm the model with information. Given our model is comparatively small and not properly fine-tuned, increasing this number could prove to be detrimental to the performance, as context would be too large and could lead to hallucinations or irrelevant answers.

Score treshold is not something we have quite considered in the pipeline. Given the physical limitations, the size of the context is something we have decided to keep constant, as letting it flactuate could lead to drasticly different results, which would be hard to analyze and compare.

Possible uses of Maximal Marginal Relevance (MMR) in the retriever scoring could be a way to test improvements in the pipeline, but we've found the similarity helped us more importantly configure and tune the database and the context itself along with the prompt design, which is the next step in our implementation.

Prompt Design and Choice of LLM

Prompt Design

Initially, we suspected that hallucinations in our model's responses were caused by an inadequate system prompt. However, after testing various prompt structures, we realized that the primary issue lay in our initial noisy database (built using broad keyword searches), which often retrieved irrelevant papers. Despite this, we observed that different prompt designs still significantly influenced the model's performance.

Early experiments with longer, highly detailed system prompts proved counterproductive. These prompts included excessive instructions, which overwhelmed the model and led to inconsistent or not contextual responses. We found that shorter prompts which emphasize contextual grounding and academic tone gave us better results.

Choice of LLM

For our project's core language model, we selected Mistral 7B, an open-source transformer-based model designed for both high performance and efficiency. Mistral 7B stands out for its ability to outperform larger models such as Llama 2 13B across a variety of benchmarks, including reasoning, mathematics, and code generation, while maintaining a much smaller parameter count [6].

The model was integrated using the Hugging Face Transformers library and configured to generate responses of up to 8,192 tokens, which is particularly valuable for academic tasks that require detailed, context-rich answers. Additionally, the code connects the model to a LangChain 'ConversationalRetrievalChain'[1], which combines conversational language modeling with document retrieval, allowing the system to answer questions based on the most relevant retrieved documents.

On-Demand Paper Fetching

The ability of using keywords for searching for context that has been extensively noted in this paper so far. In reality, the keywords in question we're only actually used to find and fetch relevant papers from the internet, which were then stored in our database. Actual querying of the context was done based on the actual question, given we've felt the question itself is a better way to find relevant context, rather than simple keywords.

Given our final implementation of the pipeline actually used a proper database that was persisted to the disk, the keyword part of it was repurposed to be used as a way to find new papers, in case the user did not feel the existing database was sufficient for the question at hand. In this case, the keywords were used to search for new papers online, which were then added to the database and used as context for the question.

While we did manually input the actual keywords themselves in our test cases, we do not expect anyone approaching this implementation to do so, at least not on a regular basis. For this purposes, we have added the keyBert [7] library to our pipeline, which is used to extract keywords from the question itself. This way, the user does not have to worry about finding the right keywords, as the model will automatically extract them from the question and use them to search for new papers online.

While this solution was not properly tested, we do expect it to work well in practice, given the simplicity of the implementation and the fact that it is based on a well-known library that has been used in many other projects. Although it is important to mention, that this does leave in room for improvement, as the library itself has tuning that is available, which could significantly improve the quality of the context being fetched.

Initial Model Analysis

For the simple demonstration of the capabilities our model, using 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, when 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, such that our agent could search for proper context, and asked it the question in connection to the keywords.

The testing agents were simply given the question, as stated above.

In testing, it has been shown that ChatGPT has given more consistent results that are more to the point, than the agent Claude; While our implementation was quite similar to it in that regard.

The responses our agent pipeline has returned 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.

[1] https://python.langchain.com/api_reference/langchain/chains/langchain.chains.conversational_retrieval.base.ConversationalRetrievalChain.html

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 is 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 model 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 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.

Discussing paper retrieval and their relevance

Results

The following results illustrate how each model performed when evaluated on academic question answering tasks. We highlight key findings from both user rankings and qualitative comparisons of the generated answers.

Model Comparison - User Study

The results, shown in Figure 1, summarize how the three models were ranked by 20 university students who participated in the survey. Most of the participants have prior knowledge in Natural Language Processing (NLP), ensuring informed evaluations across the survey questions. For detailed rankings of all survey questions, please refer to Appendix B.

The baseline model, which relies only on static data stored in our vectorstore, often received the highest rankings. It performed well especially when the question could be answered using existing knowledge. This shows that the baseline model provides a strong foundation and reliable answers for many prompts.

The extended model improves on this by adding live retrieval of recent academic papers on user demand. This model generally performed better than ChatGPT and sometimes better than the baseline. It received more top and second-place rankings, indicating that having access to up-to-date information helps improve answer quality, especially for questions requiring current research.

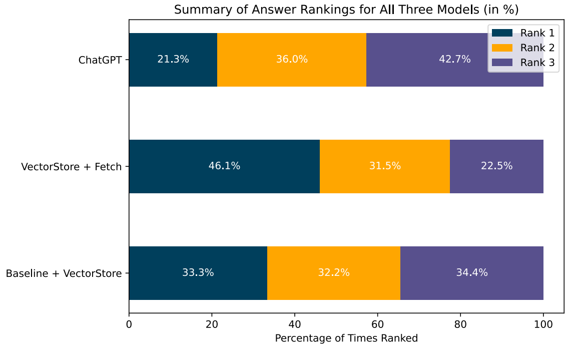

Figure 1. Overall Model Performance Based on User Rankings

Even though ChatGPT is a powerful conversational agent, its rankings were mixed in this academic context, it tended to receive more third-place votes compared to our models. It is important to note that ChatGPT's responses in this evaluation were more concise than usual due to the word count limit, which was intended to keep the comparison fair. While this model typically generates more elaborate and detailed answers, the shorter format sometimes led to responses that felt overly general or less tailored to the academic context. In contrast, our models allow users to influence the answer more directly by using keywords, which often resulted in more focused and relevant content. Although ChatGPT was limited in length, it is still expected to provide high-quality answers. However, this constraint may have contributed to its lower rankings compared to our more adaptive system.

Although the user study results show that participants mostly preferred our extended model, the model does have some shortcomings. Several limitations and potential improvements were noted, which are discussed throughout the report.

Model Comparison - Overview of Answers

Our evaluation revealed important differences in how each model (Baseline + Vectorstore, Vectorstore + Fetch, ChatGPT, and Claude) handled academic queries. Some of the asked questions are included in Appendix A.

The baseline model gave solid answers. Since it uses a database of articles from 2023 onwards, it already includes fairly recent research. However, it doesn't search for new articles beyond what's already stored. We found, it also has trouble remembering all details for certain paper, e.g. the model forgets author's name. The extended model builds on this by fetching even more papers based on keywords from the user's question or given directly from the user. This allows user to, through keywords fetch papers that are the most related to their topic of research. Still, we found that it sometimes picked articles that were very niche but not specific enough to the user's exact context, which made some answers less useful overall. We found that it would sometimes begin to hallucinate, which is understandable for project of this scope.

Furthermore, we observed inconsistencies in the formatting and structure of the models' responses. In some cases, answers were organized using numbered or bulleted lists, while in others, the same type of content was presented as plain text. There were also irregularities in the use of formatting elements such as bold text for emphasis or section headings. This lack of consistency may affect readability and user experience, particularly when comparing outputs across different queries or models. In the future, fine tuning of the answer structure is needed.

Our opinion is that while ChatGPT gives clear answers, they're often not detailed enough for research questions. Claude does better overall. It gives more complete and accurate answers while still being understandable. We found Claude's responses to be the most useful, with the most readable structure and relevant information.

In summary, the quality of the answers depends a lot on the keywords provided by the user and how detailed the prompts are. Both of our models showed solid performance overall, though there are still some issues, such as overly niche article choices or occasional hallucinations.

Discussion

The project demonstrated several notable strengths in its approach to integrating Retrieval-Augmented Generation (RAG) with targeted academic paper retrieval. One of the main successes was the ability to build a focused, up-to-date database by leveraging APIs from sources such as arXiv, Semantic Scholar, ACL, and OpenAlex. This allowed the system to provide contextually relevant and recent information, addressing a common shortcoming of general-purpose language models, which often rely on outdated training data. The use of FAISS for similarity search and the decision to limit the number of context documents helped maintain answer relevance and prevented the model from being overwhelmed with excessive or irrelevant information. Additionally, the implementation of automated keyword extraction using the keyBert library streamlined the process of fetching new papers, making the system more user-friendly and adaptable to new topics.

Despite these strengths, the project also encountered challenges. The initial method of building the database using a broad set of keywords resulted in a noisy dataset with many irrelevant articles, which negatively impacted retrieval quality. This issue highlighted the importance of careful source selection and filtering, leading to a shift toward category-based filtering and the use of more domain-specific sources like ACL, which immediately improved results. Technical limitations also posed problems, particularly with restricted access to certain scraping tools and APIs on the HPC cluster, requiring manual workarounds and additional effort to transfer data. Furthermore, the evaluation of the system was primarily subjective, and while the agent's responses were generally relevant and student-like, they sometimes lacked the completeness and depth seen in larger, more established models like Claude.

If we were to repeat this project, we would focus on three main improvements. First, we would implement a more systematic and quantitative evaluation process to better measure answer quality. Second, we would refine the template for generated answers to ensure more consistent and useful outputs. Finally, we would spend more time fine-tuning the language model itself to improve both the depth and accuracy of its responses. These changes would help address the main shortcomings we encountered and make the system more robust and effective for academic use.

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.

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.

[5] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

[6] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[7] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020.

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.

Appendix B

Figures 2 - 7 contain information on how many times (in percentages) each model was ranked 1-3 for each question from the Questionnaire.

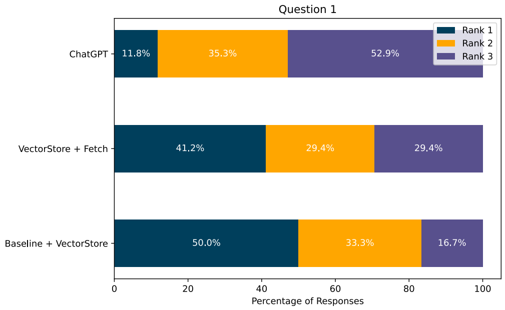

Figure 2. Answer rankings of all three models for Question 1: What is natural language processing?

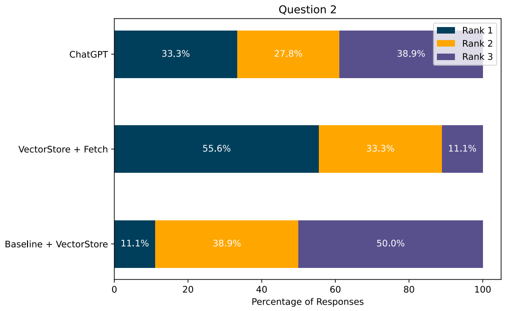

Figure 3. Answer rankings of all three models for Question 2: What are some recent papers published in Natural Language Processing domain?

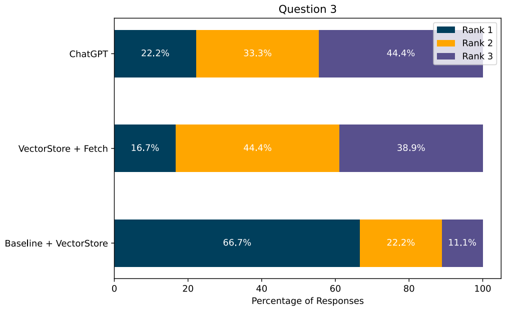

Figure 4. Answer rankings of all three models for Question 3: How would I use ML to reconstruct damaged artwork?

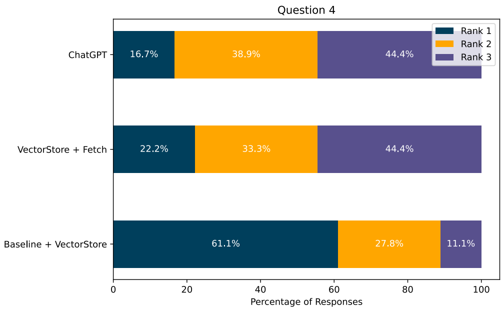

Figure 5. Answer rankings of all three models for Question 4: What are good datasets for training sentiment analysis models?

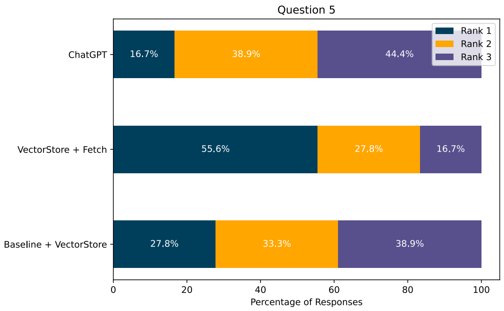

Figure 6. Answer rankings of all three models for Question 5: Tell me everything you know about unsupervised learning in NLP. Can you give me some references or links I can check out?

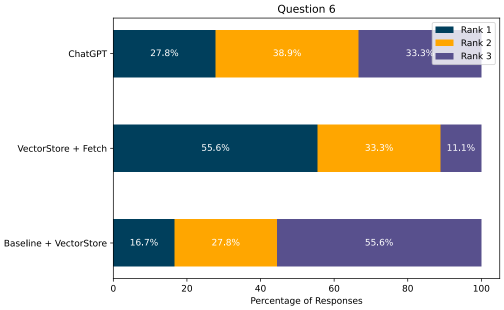

Figure 7. Answer rankings of all three models for Question 6: I want to start learning about RAG in Natural language Processing. What literature and articles do you suggest I go through? Can you give me dates of their release as well?

Appendix C

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

- **Document 1:**
  A Deeper (Autoregressive) Approach to Non-Convergent Discourse Parsing https://aclanthology.org/2023.emnlp-main.796/(Similarity: 0.7283)

- **Document 2:**
  Documenting the Unwritten Curriculum of Student Research https://aclanthology.org/2024.teachingnlp-1.1/ (Similarity: 0.7751)

- **Document 3:**
  Termout: a tool for the semi-automatic creation of term databases https://aclanthology.org/2023.contents-1.2/ (Similarity: 0.8195)

- **Document 4:**
  Teaching LLMs at Charles University: Assignments and Activities https://aclanthology.org/2024.teachingnlp-1.9/ (Similarity: 0.8218)

Question: Why are transformers important?

- **Document 1:**
  A Reasoning Paths Ranking Method for Reasoning Optimization of Small-scale Large Language Models https://aclanthology.org/2024.ccl-1.72/ (Similarity: 1.0017)

- **Document 2:**
  AutoPEFT: Automatic Configuration Search for Parameter-Efficient Fine-Tuning https://aclanthology.org/2024.tacl-1.29/ (Similarity: 1.0431)

- **Document 3:**
  Can Foundation Models Watch, Talk and Guide You Step by Step to Make a Cake? https://aclanthology.org/2023.findings-emnlp.824/ (Similarity: 1.1258)

- **Document 4:**
  Example-Driven Course Slides on Natural Language Processing Concepts https://aclanthology.org/2024.teachingnlp-1.2/ (Similarity: 1.1726)

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

- **Document 1:**
  Resolving Legalese: A Multilingual Exploration of Negation Scope Resolution in Legal Documents https://aclanthology.org/2024.lrec-main.1220/ (Similarity: 0.9462)

- **Document 2:**
  Using Machine Learning to Uncover the Semantics of Concepts: How Well Do Typicality Measures Extracted from a BERT Text Classifier Match Human Judgments of Genre Typicality? https://doi.org/https://doi.org/10.15195/v10.a3 (Similarity: 1.0578)

- **Document 3:**
  UFCNet: Unsupervised Network based on Fourier transform and Convolutional attention for Oracle Character Recognition https://aclanthology.org/2024.ml4al-1.11/ (Similarity: 1.1810)

- **Document 4:**
  A new machine-actionable corpus for ancient text restoration https://aclanthology.org/2024.ml4al-1.7/ (Similarity: 1.1844)

Question: What are good datasets for training sentiment analysis models?

- **Document 1:**
  Evidentiality-aware Retrieval for Overcoming Abstractiveness in Open-Domain Question Answering https://aclanthology.org/2024.findings-eacl.130/ (Similarity: 0.8171)

- **Document 2:**
  Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review https://doi.org/https://doi.org/10.3390/a16050236 (Similarity: 0.8523)

- **Document 3:**
  Catastrophic Forgetting in Deep Learning: A Comprehensive Taxonomy https://doi.org/https://doi.org/10.5753/jbcs.2024.3966 (Similarity: 0.8582)

- **Document 4:**
  MCECR: A Novel Dataset for Multilingual Cross-Document Event Coreference Resolution https://aclanthology.org/2024.findings-naacl.245/ (Similarity: 0.8713)