RAG with llama-2 and Mistral

January 2024

Alexander Bridgewater, Gustav Lundberg, Cecilia Nyberg, Rikard Radovac
gustavlu@chalmers.se, cecnyb@chalmers.se,
radovac@chalmers.se

Abstract

A well-known problem with large language models is the difficulties and costs associated with updating them with new or context-specific information. One way to deal with this problem is to use Retrieval-Augmented Generation (RAG) [1]. This report examines the impact of using RAG on large language models when answering context-specific questions regarding content on a website. RAG is applied to two generative models, Llama-2-13b-chat and Mistral-7b-Intruct-v.02, to answer pre-written questions about the content on a website (https://quotes.toscrape.com/page/1). The study compares the performance from these models using RAG with gpt3.5-turbo answering the same questions without RAG. The generative models were combined with two different embedding models, paraphrase-distilroberta-base-v1 and intfloat/e5-base. The result showed higher scores for exact match, F1 and semantic similarity when answering the questions using RAG compared to without, despite gpt3.5-turbo being a larger model than Llama-2 and Mistral. The best-performing setup was Mistral as the generative model and intfloat/e5-base as the embedding model. The code is available at `https://github.com/rikardradovac/RAG`

1 Introduction

Recent years have witnessed a surge in the development of generative models within the field of machine learning. Among these, generative language models have gained significant attention. These models, commonly referred to as Large Language Models (LLMs), leverage statistical methods to process extensive datasets during their *pretraining* phase. This pretraining enables LLMs to recognize and replicate textual patterns, thereby equipping them with the capability to generate contextually relevant text in response to user inputs. The designation of these models as "large" stems not only from their substantial parameter count but also from the depth and breadth of their training data. They are typically trained on large amounts of textual data available on the web, often gathered through web crawling [2]. Prominent examples of LLMs include the family of GPT, BERT, and LLama-based models, which have markedly surpassed the capabilities of their predecessors in a very short amount of time [3][4][5][6].

Despite these advancements, LLMs are not without their challenges. Of them, some concern risks such as difficulty in ensuring privacy, the use of them in frauds, and the environmental impact of the extensive computational resources [7][8]. Another notable concern is the cost associated with continually updating these models with the latest information [9]. This is particularly crucial for time-sensitive tasks such as high-frequency trading, where the integration of the most recent financial news into the model is imperative. Unfortunately, the cost of updating LLMs escalates in proportion to their size, presenting a significant hurdle given their extensive parameterization [3]. Another issue that has sparked widespread discussion is the phenomenon of *hallucinations* in LLMs [10]. These hallucinations manifest as statements that are inaccurate, misleading, or entirely fabricated, yet presented with a veneer of credibility. This occurs because LLMs, being statistical, lack an intrinsic understanding of truth or factual accuracy, leading them to occasionally generate convincing but false content.

To address these limitations, an increasing num-

ber of generative models are incorporating retrieval mechanisms, specifically through the implementation of retrieval heads. These enhanced models, known as Retrieval Augmented Generative (RAG) models, utilize a retrieval head to sift through their extensive knowledge bases, identifying text that is most relevant to a given prompt [9]. This process typically involves a similarity assessment, often using measures like Cosine similarity, to gauge the relevance between the user's prompt and the information available in the knowledge base [11]. Unlike traditional models that primarily rely on grammatical patterns and pre-trained data, RAG models focus on fetching factual content [9]. This content, predominantly comprising factual descriptions of events, can be dynamically added to the knowledge base, allowing for continual expansion and update of the information repository. Such a feature is particularly advantageous as it ensures that the model can access and provide information that was not included in its initial training phase [12]. This capability to retrieve and integrate up-to-date factual data in real-time has significantly enhanced the utility and applicability of these models. They have become invaluable assets in fields where accuracy and timeliness of information are crucial, effectively bridging the gap between static pre-training and the dynamic nature of real-world information.

1.1 Problem statement

The purpose of this report is to investigate the impact of integrating Retrieval-Augmented Generation (RAG) into large language models for context-specific question answering.

However, to examine this impact, there are several aspects to consider. In this report, the focus is on quantitatively measuring if the incorporation of RAG on language models can lead to a statistically significant increase in the overall accuracy of question answering concerning content on a website. Specifically, the investigation aims to determine if a smaller model using RAG can outperform a larger model not using RAG.

Another important aspect to consider when investigating if a smaller model can outperform a larger one centers on the exploration of how various embedding models affect performance when integrated with the RAG model for context-specific question answering. This analysis aims to identify the nuances in performance and interaction between embedding models and RAG, contributing to the optimization of model combinations for improved effectiveness.

Furthermore, an additional influencing factor is the impact of the generative model on the accuracy of context-specific question answering when used in conjunction with RAG. The purpose of this is to find insights into different models' suitability for context-specific question-answering scenarios when integrated with RAG, to find the best-performing combination.

This culminates in the formulation of the following three research questions:

Research Questions

- Can the incorporation of RAG lead to a statistically significant increase in the overall accuracy of question answering regarding specific content on a website?
- How do different embedding models affect the performance when integrated with the RAG model for specific question answering?
- How do different generative models impact the accuracy of specific question answering when used in conjunction with the RAG model?

1.2 Limitations

- Only two different embedding models and two different generative models will be used.
- Only one website was used to measure the performance of RAG.

2 Theory

This section briefly presents the theory behind transformers, attention, the encoder and decoder, sentence embeddings, the rag-model, quantization as well as the evaluation metrics later used in the report.

2.1 Transformers

The Transformer is a neural network architecture used in the field of sequence-to-sequence learning [13]. Unlike traditional models that rely on recurrent neural networks (RNNs) and convolutional neural networks (CNNs), the transformer is built solely on attention mechanisms, allowing it to capture dependencies across input and output sequences without the need for sequential processing.

The Transformer is designed with a structure that uses identical stacks of layers, where the input sequences are processed through the encoder, and the output sequences are generated autonomously by the decoder [13]. The architecture is shown in Figure 1. The innovation at its core revolves around the attention mechanism, specifically adopting the multi-head attention approach.

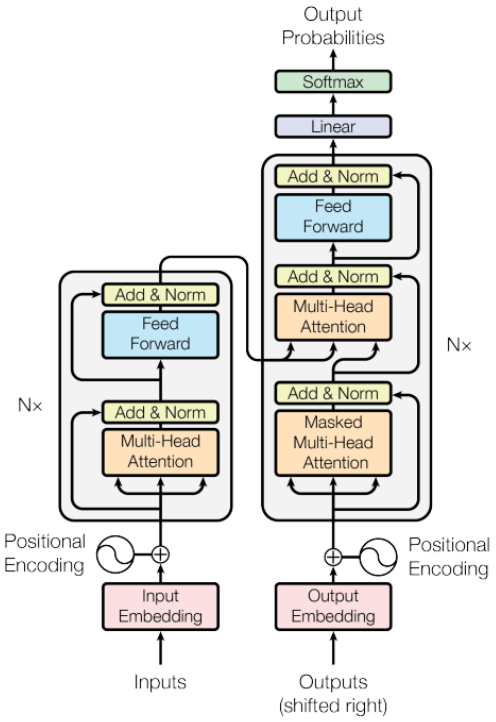


Figure 1: Example of a figure.

The attention mechanism, responsible for mapping queries and key-value pairs to an output, involves weighted sums of values based on the compatibility between queries and keys [13]. The Transformer adopts the "Scaled Dot-Product Attention" mechanism as seen in 2, which enables the model to weigh the importance of different elements in the input sequence when generating each element in the output sequence.

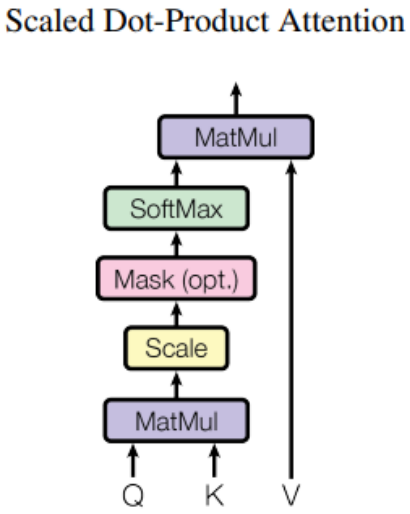


Figure 2: Scaled Dot-Product Attention.

The Multi-Head Attention as seen in Figure 3 further elevates the model's capabilities by projecting queries, keys, and values into distinct subspaces, performing parallel attention, and merging the outcomes [13]. This method empowers the model to collectively focus on information across varied representation subspaces, thereby enhancing overall performance.

2.2 Sentence Embeddings

Sentence embedding is a technique in Natural Language Processing (NLP) that represents sentences as fixed-length numerical vectors, allowing computers to comprehend and process their semantic meaning [14]. This allows textual analysis of cosine similarity, which quantifies the similarity between two vectors within an inner product space [15]. The score is determined by whether the angle between the vectors are aligned in the same direction. Unlike word embedding, which focuses on individual words, sentence embedding condenses entire sentences into a single vector.

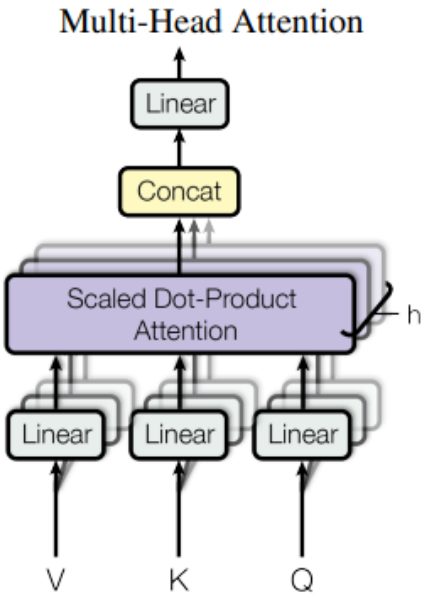


Figure 3: Example of Multi-Head Attention. [13]

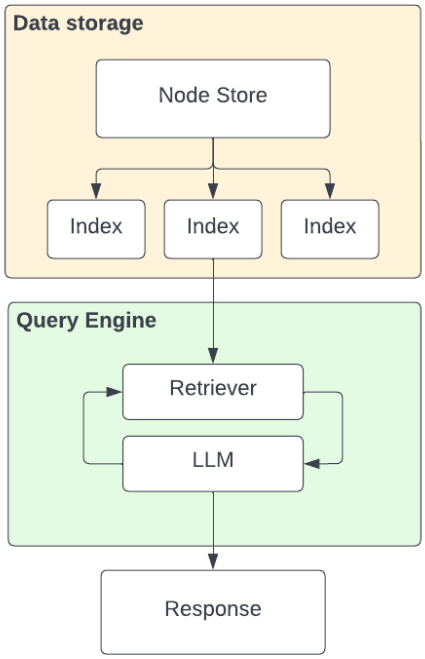


Figure 4: Example of Retrieval Augmented Generation (RAG)

2.3 RAG

The Retrieval-Augmented Generation (RAG) model represents an architecture that integrate pre-trained parametric and non-parametric memory to promote language generation capabilities [1]. In its structure, the pre-trained sequence-to-sequence model uses parametric memory, with non-parametric memory relying on a dense vector index accessed through a pre-trained neural retriever.

The retriever uses cosine similarity scores to retrieve relevant information from the database, where text embeddings are stored as indices. This retrieval process is initiated by comparing the initial query or question to all entries in the database. The top k entries, determined by their cosine similarity scores, are then integrated into the language model as context. An outline for the architecture is depicted in Figure 4. The provided context is then used to supposedly generate a more suitable answer to the query in question.

2.4 Evaluation Metrics

The foundation for evaluating a question-answering (QA) system lies in the metrics associated with the reader and retrievers [16].The contrast in functions between the reader and retriever emphasizes the need for distinct evaluation metrics. When evaluating the reader node, the retrieval process is overlooked, and the reader receives the document containing the answer span directly. The evaluation can then be carried out using various metrics such as Exact match, F1 Score, Semantic Answer Similarity etc.

The exact match (EM) is used as an exact metric, examining the precision of the predicted answer by requiring an identical match with the correct answer. For example, even a slight deviation in wording results in a zero EM score. However, the F1 score takes a more tolerant approach, measuring word overlap between predicted and labeled answers. This metric is more forgiving regarding the similarity of two answer strings.

Semantic Answer Similarity (SAS) stands out in the evaluation of QA systems, as it mitigastes the constraints associated with conventional metrics. Unlike F1 and EM, SAS uses a Transformer-based cross-encoder architecture to assess semantic similarity between answers, disregarding lexical overlap. This inovation allows for a more comprehensive evaluation that captures equivalence between answers that may differ in tokens. The use of SAS in the evaluation process enhances the system's ability to understand and respond to user queries in a more semantically meaningful way.

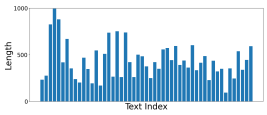
3 Method

The method section outlines the steps taken to create the material needed to analyze the performance of question answering using RAG. These steps include the initial data collection and preprocessing stages, the model architecture selection, and evaluation metrics.

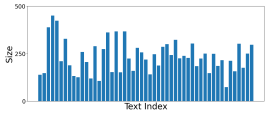
3.1 Data

The data that the RAG algorithm retrieves from was scraped from the website https://quotes.toscrape.com/page/1/. This website contains quotes from, and information about, famous authors. The choice to use this website was because the content was easy to access and it was possible to create specific questions that should be easier to answer for a model with access to this exact information. This data was embedded in two different ways, firstly using the sentence transformer paraphrase-distilroberta-base-v1 and secondly using intfloat/e5-base. The embedded text was then stored in Pinecone.

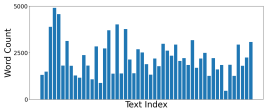
To be able to evaluate how well the models perform, two test sets were made with questions and answers from the scraped website. The first test set is made up of 102 questions and the second is made up of 233 questions. These questions were made with the intention of them being easier to answer for a model with specific knowledge about the content on the web page and harder for a model with just general knowledge. In table 1, the first five questions from the second test set are shown.



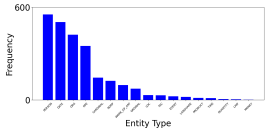
(a) Average word lengths per document



(b) Vocalbulary size per document



(c) Character count per document



(d) Named entity distribution for all documents

Figure 5: Data analysis for the texts

Furthermore, to provide a comprehensive insight into the data we performed data analysis showcased in Figure 5. We chose to analyze vocabulary size, average word lengths, word count, and named entity types to gauge the complexity of the documents, as well as how they differ from each other. In this case, the distributions tell the story of a homogenous set of documents set of documents with low deviations in characteristics among documents. Moreover, the entity types that are the main subject of the question while quizzing the RAG, belong mainly to real-world entities such as persons, dates, and organizations, rather than more abstract concepts such as events or ideas.

Q & A
Who said, "The world as we have created it is a process of our thinking"? Albert Einstein
According to J.K. Rowling, what reveals what we truly are? Choices
How does Albert Einstein suggest one can live their life? Two ways: as though nothing is a miracle or as though everything is a miracle
What does Marilyn Monroe consider beauty and genius? Imperfection, madness, and being absolutely ridiculous
According to Albert Einstein, what should one aim to become rather than a man of success? Man of value

Table 1: Famous Quotes and Their Authors, data2

3.2 RAG

Two different models were used for answering the questions with RAG, Llama-2-13b-chat-hf and Mistral-7B-Instruct-v0.2. These two models were chosen because they had relatively high MT-bench scores and fewer parameters than gpt-3.5-turbo which was chosen as a larger LLM to compare the results with.

To determine which prompt to send to the models, different prompts were tried out on chatGPT. The goal with the prompt was to make the model answer the questions in a similar way to the answers in the test sets. These answers are short and concise, often consisting of only one word, intending to make the evaluation easier. A problem encountered here was that chatGPT often answered with unnecessary words, for example by repeating the question in the answer. On the other hand, if

Prompt Templates
Llama Template
`[INST] <<SYS>> Answer the question based on the context below. Keep the answer short and concise. Respond "Unsure about answer" if not sure about the answer. Do not provide any prefix, simply write out the answer. {context}. <</SYS>> question: {question}[/INST]`
Mistral Template
`<s>[INST] Answer the question based on the context below. Keep the answer short and concise. Respond "Unsure about answer" if not sure about the answer. Do not provide any prefix, simply write out the answer. {context}. question: {question}[/INST]`
OpenAI Template
`You are a helpful assistant. You answer questions in a short, concise and informative way. Respond "Unsure about answer" if not sure about the answer. Do not provide any prefix, simply write out the answer. question: {question}`

Table 2: Prompt Templates

given a too strict prompt, for example to answer as short as possible, some important information was left out. Therefore, the prompt could not be too strict nor to loose about the length of the answers. The prompt also needed to include something about what the model should do if it was not sure about the answer, to avoid hallucinations. The resulting prompts can be seen in table 2.

3.3 Evaluation metrics

Three different evaluation metrics wtheysed, to get a more comprehensive idea of the models' performances. These evaluation metrics were exact match, F1 scores, and semantic similarity scores(STS). An independent embedding model was used to calculate the similarity scores to keep the results unbiased. The model used was sf_model_e5, since this model had a high STS rank [17].

In the evaluation of embedding models, they were compared using the same generative model separately for Llama2 and Mistral. This was done respectively on both the two different datasets.

The evaluation of generative models consisted of comparing the two embedding models paired with the Mistral and comparing it to the two embedding models paired with the Llama2. Each comparison was done on both two datasets. An overall best GPT and embedding model could then be determined as an overall best model according to the evaluation metrics.

The overall best model could then be used to be compared to an even bigger LLM for comparison of the use of Retrieval Augmented Generation and no RAG.

4 Result

The results of the runs are presented in this section. A more extensive display of results can be seen in appendix.

Comparing the embedding models			
Mistral-7B	Exact match	F1 score	Similarity Score
e5 and data1	0.3786	0.5656	0.7299
e5 and data2	0.2886	0.5672	0.7543
paraphrase data1	0.3286	0.4957	0.7179
paraphrase data2	0.1000	0.4514	0.6810

Table 3: Embedding models compared on Mistral7b

Comparing the embedding models			
Llama-2	Exact match	F1 score	Similarity Score
e5 and data1	0.2943	0.4858	0.6638
e5 and data2	0.2357	0.4834	0.6647
paraphrase data1	0.2429	0.3843	0.6396
paraphrase data2	0.1143	0.4424	0.6574

Table 4: Embedding models compared on Llama2

Comparing the GPT models			
Mistral-7B	Exact match	F1 score	Similarity Score
e5 and data1	0.3786	0.5656	0.7299
e5 and data2	0.2886	0.5672	0.7543
paraphrase data1	0.3286	0.4957	0.7179
paraphrase data2	0.1000	0.4514	0.6810
Llama-2	Exact match	F1 score	Similarity Score
e5 and data1	0.2943	0.4858	0.6638
e5 and data2	0.2357	0.4834	0.6647
paraphrase data1	0.2429	0.3843	0.6396
paraphrase data2	0.1143	0.4424	0.6574

Table 5: Scores for Mistral and Llama2 on the two embedding models

RAG vs NO RAG			
Mistral-7B	Exact match	F1 score	Similarity score
e5 and data1	0.3786	0.5656	0.7299
e5 and data2	0.2886	0.5672	0.7543
3.5 TURBO	Exact match	F1 score	Similarity Score
data1	0.0089	0.2135	0.4129
data2	0.0030	0.1546	0.3310

Table 6: Scores for Mistral with RAG paired with e5 compared to OpenAi ChatGPT 3.5 Turbo without RAG

5 Discussion

Evaluating various models, such as embedding models and generative models, including the comparison between using and not using Retrieval-Augmented Generation (RAG), provides valuable insights into the effectiveness of different configurations.

In the evaluation of the embedding models, as seen in Table 3 and Table 4, the intfloate5-base consistently outperforms the paraphrase-distilroberta-base-v1 in terms of exact matches, F1 scores, and similarity scores, regardless of the generative model used. This indicates that the intfloate5-base sentence transformer is more proficient at capturing the nuances and style of the provided quotes and is capable of reproducing more contextually accurate and coherent responses. This shows the importance the selection of an embedding model can have in the predictive capabilities of different language models.

In the comparison of generative models, as seen in Table 5, Mistral consistently emerges as the more effective model across various datasets and embedding models, showcasing superior performance in terms of exact matches, F1 scores, and similarity scores when compared to Llama2. This superiority could be attributed to Mistral's architectural nuances and possibly its training data, enabling it to better understand and reproduce the complexity of language specific to the domain of famous quotes.

Among the models examined, the standout performer is the setup that uses the intfloate5-base sentence transformer model paired with the Mistral-7B generative model. This model consistently achieves the highest scores in terms of all metrics, demonstrating the significant impact of combining a generative model, specific embedding choices, and the benefits of RAG in generating data-based responses.

In terms of generative versus retrieval-augmented approaches, the findings suggest that models that incorporate retrieval mechanisms outshine general generative models. This is evident in the comparison with ChatGPT, depicted in Table 6, where the RAG models consistently deliver much higher performance in terms of Exact Matches, F1 Scores, and Similarity Scores, despite the fact that the model without RAG is a much bigger LLM in terms of parameters.

As we prefaced in the method, the data is homogenous. Thus, the results obtained may not ring true for all types of data. Especially not when considering different context window sizes for the RAG model. Furthermore, it contained an overwhelming amount of relatively easy entities to identify, which may impact the results when its based on more abstract concepts or ideas. How-

ever, we it is hard to conclude how difficult these reasoning steps for these entities were.

Scoring and comparing answers written in text can be difficult since several answers may be correct but expressed in different ways. The answers shown in table 7 are examples of answers that are easy to score since they perfectly align with the manually created correct answer. All answers in table 7 receive perfect scores on all evaluation metrics. This was, however, not what the majority of the generated answers looked like.

```
Question: What awards have Suzanne
Collins' Hunger Games books won?

llama 2 paraphrase - The Hunger Games
trilogy by Suzanne Collins has won
several awards, including the GA
Peach Award.

True answer: The Hunger Games trilogy
by Suzanne Collins has won several
awards, including the GA Peach Award.

Question: What, according to
Friedrich Nietzsche, makes unhappy
marriages?

llama 2 e5 - Lack of friendship.
llama2 paraphrase - Lack of
friendship.
mistral 7b paraphrase - Lack of
friendship.
mistral 7b e5 - Lack of friendship.

True answer: Lack of friendship.
```

Table 7: Example of perfect answers

Table 8 shows an example where ChatGPT is hallucinating an incorrect answer while Mistral generates an accurate answer but with fewer words than the correct one. Even though ChatGPT's answer is wrong and Mistral's is right, the F1 score for ChatGPT is probably not that much lower than for Mistral. This is because Mistral did not include the word missions in its answer, which was included in the true answer. ChatGPT did include missions but had an incorrect number, which was the most important part of the answer from a semantic perspective, but this is not taken into consideration for the F1 score.

There were also cases where some additional information, correct but unnecessary, was added to the answer, resulting in a lower score.

Therefore, the scores for the different models are highly dependent on the text format and word

```
Question: How many missions were
operated by Mother Teresa's
Missionaries of Charity at the time
of her death?

Mistral 7b - e5-base output: 610

ChatGPT: Over 4,000 missions.

True answer: 610 missions
```

Table 8: Example of incorrect answer

choices used when generating the answers, potentially leading to the results not completely reflecting which setup is truly the best at producing logically correct answers. Using human evaluation could solve this problem, but since there were almost 2000 answers to score, this would not have been possible. Another way to deal with this is to try using different prompts and see how the results differ since the prompt highly affects the outlook of the generated answers. The prompt can affect how long the answers are, how formal the language is, what style is used, and much more. It is possible that the result presented in this report would look different if different prompts were used. However, since the semantic similarity score reflects the underlying logical meaning of the answer, this metric would probably change the least with different prompts. This is an indicator that the best-performing setup according to the semantic similarity score would potentially be the same for other prompts as well, but to determine this, more experiments would have to be made.

Another aspect that may impact the accuracy of the results is that RAG is used on one kind of model and then compared with the results from another kind of model. It could potentially have been easier to answer the problem statement if the scores for Llama2 and Mistral answering the questions with and without RAG were compared instead of comparing these results with OpenAI. This approach would help isolate the effects of RAG and minimize confounding factors introduced by comparing results across different models. On the other hand, the comparison between a model with more parameters not using RAG and fewer parameters using RAG would have been lost.

6 Conclusion

The results in this report support the idea that using RAG for question answering increases the accuracy of the answers, making it possible for a language model with fewer parameters to outperform a larger one. The findings also indicate that the choice of embedding model and generative model to use RAG on has a notable impact on the performance. When using exact match, F1 score, and similarity score, the best-performing setup used intfloat5 as the embedding model and Mistral-7B-Instruct-v0.2 as the generative model. It is possible that the outcome would differ if another prompt was used or if the answers were evaluated by humans.

7 Future work

Some possible future work inspired by this project includes:

- Measuring the effects of different prompts by repeating the experiments and comparing the results.
- Repeating the experiments but using the same generative model with and without RAG. This would make it possible to explore the effect RAG has on hallucinations.
- Using a different dataset, for example, one with time-sensitive information.

References

[1] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," May 2020. DOI: https://doi.org/10.48550/arXiv.2005.11401.

[2] X. Lareo, *Large language models (llm)*, https://edps.europa.eu/data-protection/technology-monitoring/techsonar/large-language-models-llm_en, [Online; accessed 14-January-2024], 2024.

[3] Y. Chang, X. Wang, J. Wang, *et al.*, *A survey on evaluation of large language models*, 2023. arXiv: 2307.03109 [cs.CL].

[4] H. Touvron, T. Lavril, G. Izacard, *et al.*, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL].

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. [Online]. Available: https://arxiv.org/abs/1810.04805.

[6] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

[7] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '21, Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623, ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. [Online]. Available: https://doi.org/10.1145/3442188.3445922.

[8] L. Weidinger, J. Uesato, M. Rauh, *et al.*, "Taxonomy of risks posed by language models," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22, Seoul, Republic of Korea: Association for Computing Machinery, 2022, pp. 214–229, ISBN: 9781450393522. DOI: 10.1145/3531146.3533088. [Online]. Available: https://doi.org/10.1145/3531146.3533088.

[9] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 9459–9474. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

[10] V. Rawte, A. Sheth, and A. Das, *A survey of hallucination in large foundation models*, 2023. arXiv: 2309.05922 [cs.AI].

[11] A. F. Smeaton, "Using nlp or nlp resources for information retrieval tasks," in *Natural Language Information Retrieval*, T. Strzalkowski, Ed. Dordrecht: Springer Netherlands, 1999, pp. 99–111, ISBN: 978-94-017-2388-6. DOI: 10.1007/978-94-017-2388-6_4. [Online]. Available: https://doi.org/10.1007/978-94-017-2388-6_4.

[12] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, *A survey on retrieval-augmented text generation*, 2022. arXiv: 2202.01110 [cs.CL].

[13] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention Is All You Need*, Aug. 2023. DOI: https://doi.org/10.48550/arXiv.1706.03762. (visited on 01/12/2024).

[14] N. Reimers and I. Gurevych, *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*, Aug. 2019. DOI: https://doi.org/10.48550/arXiv.1908.10084. (visited on 01/12/2024).

[15] *Cosine Similarity - an overview — ScienceDirect Topics*, Jan. 2024. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/cosine-similarity (visited on 01/12/2024).

[16] *Metrics to Evaluate a Question Answering System — deepset*, Sep. 2021. [Online]. Available: https://www.deepset.ai/blog/metrics-to-evaluate-a-question-answering-system (visited on 01/12/2024).

[17] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "Mteb: Massive text embedding benchmark," *arXiv preprint arXiv:2210.07316*, 2022. DOI: 10.48550/ARXIV.2210.07316. [Online]. Available: https://arxiv.org/abs/2210.07316.

Appendix

Exact Matches	F1 Scores	Similarity Scores
0.294 286	0.485 829	0.663 799

Table 9: intfloate5-base_meta-llamaLlama-2-13b-chat-hf_rag_data.csv

Exact Matches	F1 Scores	Similarity Scores
0.235 714	0.483 445	0.664 694

Table 10: intfloate5-base_meta-llamaLlama-2-13b-chat-hf_rag_data2.csv

Exact Matches	F1 Scores	Similarity Scores
0.378 571	0.565 584	0.729 876

Table 11: intfloate5-base_mistralaiMistral-7B-Instruct-v0.2_rag_data.csv

Exact Matches	F1 Scores	Similarity Scores
0.288 571	0.567 176	0.754 293

Table 12: intfloate5-base_mistralaiMistral-7B-Instruct-v0.2_rag_data2.csv

Exact Matches	F1 Scores	Similarity Scores
0.242 857	0.384 272	0.639 630

Table 13: paraphrase-distilroberta-base-v1_meta-llamaLlama-2-13b-chat-hf_rag_data.csv

Exact Matches	F1 Scores	Similarity Scores
0.114 286	0.442 400	0.657 408

Table 14: paraphrase-distilroberta-base-v1_meta-llamaLlama-2-13b-chat-hf_rag_data2.csv

Exact Matches	F1 Scores	Similarity Scores
0.328 571	0.495 678	0.717 858

Table 15: paraphrase-distilroberta-base-v1_mistralaiMistral-7B-Instruct-v0.2_rag_data.csv

Exact Matches	F1 Scores	Similarity Scores
0.100 000	0.451 381	0.680 959

Table 16: paraphrase-distilroberta-base-v1_mistralaiMistral-7B-Instruct-v0.2_rag_data2.csv

Exact Matches	F1 Scores	Similarity Scores
0.008 911	0.213 458	0.412 913

Table 17: rag_data.csv_openai_predictions

Exact Matches	F1 Scores	Similarity Scores
0.003 017	0.154 552	0.331 033

Table 18: rag_data2.csv_openai_predictions