Analysing the potential solutions to LLM hallucinations in abstractive text summarisation

Abstract

This work explores the implementation of Retrieval-Augmented Generation (RAG) as a method to mitigate hallucinations in abstractive text summarization using transformer-based BART and T5. The research utilizes the HaDeS dataset to evaluate the effectiveness of RAG in improving the quality and factual consistency of generated summaries. Quantitative analysis shows that summaries generated with RAG consistently outperform those without RAG across various metrics, including ROUGE, METEOR, BERTScore, and MoverScore. Notably, BART with RAG demonstrated a 21% improvement over its non-RAG counterpart, while T5 with RAG showed a 17.3% improvement. Additionally, the study adopts a two-step summarization technique that further enhances summary quality, leading to a cumulative improvement of 16.7% when combined with optimized retrieval processes by tuning the Top_k value. Despite these advancements, manual evaluation of hallucinations suggests that while BART with RAG excels in quantitative metrics, T5 with RAG may better manage hallucinations in certain contexts. The findings underscore the potential of RAG as a robust solution for enhancing abstractive summarization compared to other approaches on their own, while also highlighting areas for future research, including the refinement of retrieval processes and the application of these techniques to larger and more diverse datasets including long text summarisation.

Introduction

In the field of Natural Language Processing (NLP), text summarization is crucial in condensing large volumes of information into manageable summaries while preserving the original text's essential meaning. As the volume of available information continues to grow exponentially, the development of automatic text summarization algorithms has become increasingly important. These algorithms have evolved significantly since Luhn's pioneering work in the late 1950s (Luhn,1958). In recent years, transformer-based large language models (LLMs) have driven substantial improvements in automatic text summarization (Raffel et al. 2020).

Text summarization techniques can be broadly categorized into extractive and abstractive approaches. While extractive summarization focuses on selecting key sentences from the source text, abstractive summarization generates new sentences that encapsulate the core meaning of the original content. The trend toward using LLMs for abstractive

summarization has gained momentum due to their ability to generate fluent and coherent summaries. However, one significant challenge associated with abstractive summarization by LLMs is the occurrence of hallucinations. Hallucination in LLM generated summaries are instances wherein the model generates information that is not present in the source material (Bruno et al. 2023).

Hallucinations can be extrinsic or intrinsic. Intrinsic hallucinations occur in a summary when the model generates text that contradicts the source text. Extrinsic hallucinations, on the other hand, are generated content that cannot be proven or disproven based on the source text. Extrinsic hallucinations can be considered non-factual when they contain information that does not align with real-world knowledge. On the other hand, extrinsic hallucinations can be factual hallucinations when they provide information matching real world knowledge which can be beneficial to summaries by providing more context to our understanding of the generated summaries (Cao M. et al., 2022).

This research aims to explore and analyze potential solutions to hallucinations in abstractive summarization. The study will utilize the HaDeS dataset and investigate various approaches previously employed by researchers, particularly the Retrieval-Augmented Generation (RAG) technique, to develop effective strategies for mitigating hallucinations in LLMs during abstractive text summarization. The text-to-text transfer transformer (T5) and BART models will serve as the LLMs for summarization in this study.

Aims and Objectives

This study aims to create a system that effectively mitigates hallucinations in abstractive text summarization. The specific objectives include:

- Developing a RAG system to support LLMs in generating accurate summaries.
- Investigating the role of retrieved semantic similarity in enhancing the quality of generated summaries.
- Examining metrics for evaluating hallucinations in LLM-generated summaries.

Significance of the study

Advancements in sequence-to-sequence architectures have enabled LLMs to generate text that rivals human fluency and coherence (Maynez et al. 2020), and in some instances, even surpass human experts in specific tasks (Wang, Y. et al. 2024), such as generation of hundreds of coherent news article or product descriptions in minutes whereas experts in the field will need more time to achieve such feats. In Natural Language Processing (NLP),

coherence refers to the way textual units relate to one another, ensuring that the generated text logically presents a clear progression of ideas (Maimon and Tsarfaty 2024). This enhanced coherence has broadened the application of LLMs across various industries. However, the potential for errors in LLM-generated summaries presents significant risks, particularly in high-stakes domains where inaccuracies can have severe consequences, such as misdiagnoses in medicine or flawed legal arguments. The ethical implications of such errors further underscore the need for reliable summarization systems, as unchecked hallucinations could lead to mistrust and resistance from users in critical sectors (Bach et al. 2024).

Literature Review

The existence of hallucinations in LLM-generated text has been well-documented by scholars (Rawte et al. 2023; Tonmoy et al. 2024; Luo et al. 2024). Abstractive summarization, which generates new tokens and phrases, is particularly susceptible to hallucinations compared to extractive summarization (Cao Z. et al. 2018). Hallucinations can be categorized as intrinsic (manipulating information not present in the source text) or extrinsic (introducing information not directly contained in the source text) (Cao M. et al. 2022).

LLMs are expected to continue hallucinating due to limitations in training data and their inference-based text generation process (Xu et al. 2024). Other causes include biased data, lack of real-world knowledge (Bruno et al. 2023), the softmax bottleneck (Chang and McCallum 2022), and diluted attention in longer sequences (Hahn 2020).

Researchers have explored various strategies to mitigate hallucinations, including Retrieval-Augmented Generation (RAG). RAG combines the power of LLMs with information retrieval from external knowledge bases (Shuster et al. 2021; Izacard & Grave 2021; Lewis et al. 2021). Other approaches include using knowledge graphs (Dziri et al. 2021; Das et al. 2022), Named Entity Recognition (NER) (Nan et al. 2021), and fine-tuning LLMs to update their knowledge base and reduce hallucinations (Tonmoy et al. 2024).

BART (Lewis et al. 2019) and T5 (Raffel et al. 2020) are particularly suitable for text summarization. While BART has demonstrated effectiveness in recovering original texts from corrupted versions, T5 has shown versatility across NLP downstream tasks (Cho et al. 2021). Embeddings improve the quality of generated outputs (Agrawal et al. 2023), and the retrieval process uses the dot product of the passage vectors as the similarity measure to identify the top-k passages most relevant for a summary, thereby reducing hallucination in the generated texts (Karpukhin et al. 2020).

Methods like factual consistency checks (Kryscinski et al. 2020) and the use of QA approaches (Wang, A., et al, 2020) and ACUEVAL which perform two fine-grained and structured sub-tasks as against asking the model to check themselves by producing a single score (Wan et al., 2024) appear to better tools as they are more interpretable than standard evaluation metrics like ROUGE, METEOR, BERTScore and MoverScore.

While RAG has been hailed as the main technique in minimizing LLM hallucination (Wu et al. 2023) in many NLP downstream tasks, including text summarization, available literature has not explored how retrieved semantic similarity of source text from a vector database can help LLMs improve abstractive summarization. This work aims to fill this gap in the corpus of literature on computational linguistics.

Approaches to Mitigating Hallucinations

There are several approaches to mitigating hallucinations in LLMs. One of these is fine-tuning which enhances the accuracy of LLMs by training them on new datasets or task-specific datasets. This process allows the model to specialize in a particular domain or update its knowledge, leading to improved performance and a reduction in hallucinations. However, a lot of fine-tuning data is required for this process, and this can be resource-intensive both in terms of time and computational power (Tonmoy et al. 2024; Liu et al. 2021).

Secondly, some researchers (Nan et al. 2021; Su et al. 2024) have proposed using Named Entity Recognition (NER) techniques. NER extracts key entities from the source text and ensures that these entities are accurately represented in the generated summary, while also identifying entities in the summary that are not present in the source text. This approach enhances factual accuracy and helps reduce hallucinations. However, NER is limited by its inability to detect non-entity factual errors.

On the other hand, the use of Knowledge Graphs (KGs) to mitigate LLMs hallucinations is gaining attention (Dziri et al. 2021; Das et al. 2022). KG provides a structured representation of entities and their relationships, which can be integrated into LLMs to mitigate both entity and non-entity hallucinations. While KGs are effective, they are expensive to construct and maintain, and their accuracy depends on the quality of the underlying data.

Notwithstanding the above approaches, the use of Retrieval-Augmented Generation (RAG) in mitigating hallucination in LLMs has dominated current approaches in the enterprise. RAG mitigates hallucinations by retrieving relevant information from an external knowledge

base during the text generation process, thereby grounding the output in factual data (Izacard & Grave 2021). The principal strength of RAG is that it combines information retrieval with text generation, pulling in real-time external data to enhance factual consistency in LLM outputs. RAG is more effective than static knowledge graphs by dynamically integrating current information without the high costs or need for expert-built knowledge bases. Unlike NER, which only handles entity-level accuracy, RAG also addresses non-entity hallucinations, making it a more comprehensive and cost-effective solution.

Furthermore, there are instances when a combination of techniques, such as integrating RAG with knowledge graphs or NER, may be necessary to effectively mitigate hallucinations. Hybrid approaches leverage the strengths of multiple methods to provide a more robust solution to the problem of hallucination in LLM-generated summaries (Edge et al. 2024).

Dataset

The dataset used for this research is HaDeS, and the name is coined from HAllucination DEtection dataSet. HaDeS is a token-level, reference-free hallucination detection dataset created from text segments extracted from English Wikipedia. The raw text was first perturbed and then verified with crowd-sourced annotations (Liu et al. 2022). The dataset, sourced from Github (www.github.com), is 10MB in size and is split into training, validation, and test sets with sizes of 8754, 1000, and 1200 samples, respectively; and is suitable for hallucination checks in summaries.

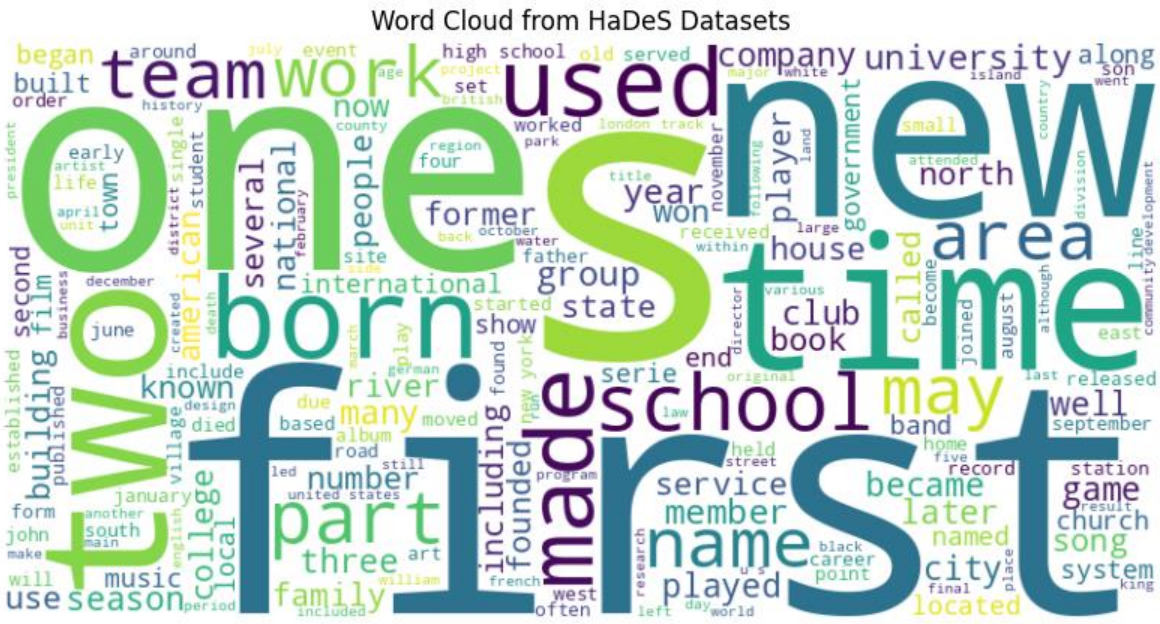

Figure 1: WordCloud from HaDeS dataset showing the distribution of words in the dataset

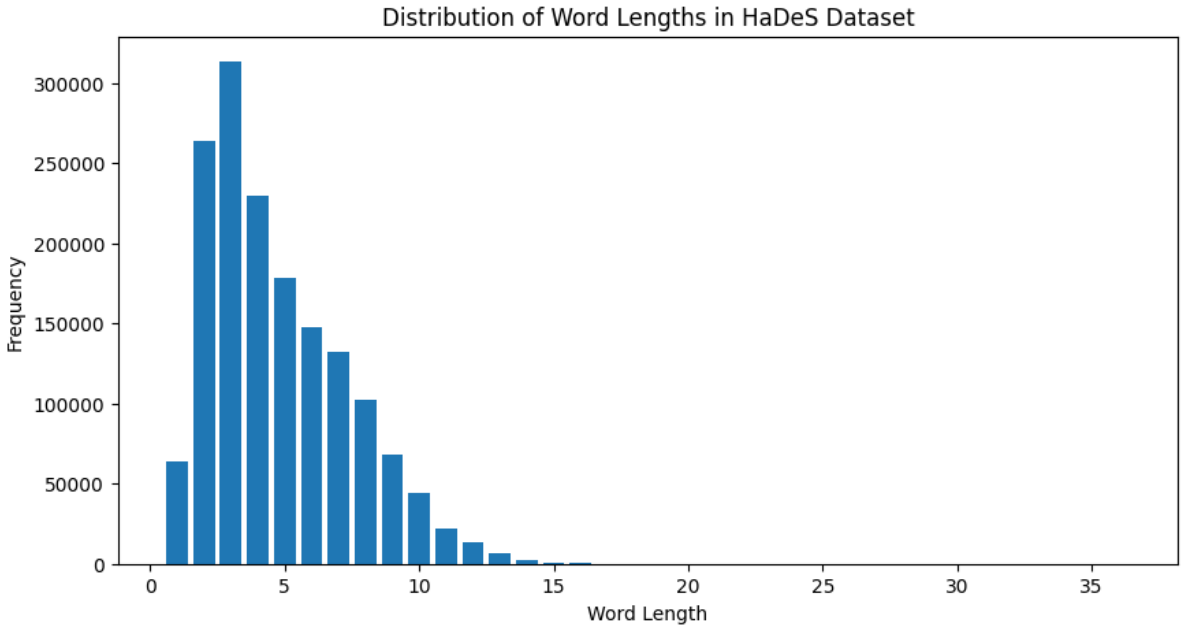

Figure 2: Distribution of word length count in HaDeS dataset

Data Preprocessing and Embedding

Initial data exploration revealed that HaDeS is a carefully curated dataset with all letters converted to lowercase. The dataset was preprocessed by cleaning the text, removing special characters (including specific sequences like "==="), and managing spaces by trimming leading and trailing spaces while reducing multiple spaces within texts to a single

space. This preprocessing step is crucial to ensure the text is in a consistent format for both training and evaluation.

Transformer-based models like BART and T5 bypass the need for traditional NLP techniques like lemmatization and stopword removal by utilizing the full input sequence in their attention mechanisms. They also produce fixed-size vectors, making dimensionality reduction unnecessary.

In addition, Universal Sentence Embedding (USE) was employed to generate embeddings from the source text, which were then stored in a Pinecone cloud-based database for efficient retrieval during the summarization process. USE has been shown to outperform word embeddings, which struggle to capture the semantics of larger text chunks (Deepthi and Sowjanya, 2021).

Data Augmentation

Random facts unrelated to the core content of the text were injected into the source text. A total of ten sentences averaging nine words per sentence were sourced from an ancient Greek mythology on Hades. Three random facts (sentences) were injected into each row. The injection of random facts was done to test the model's ability to generate coherent summaries despite the presence of irrelevant or misleading information.

Method

This study employed a comprehensive approach to evaluate and compare Retrieval-Augmented Generation (RAG) and non-RAG implementations for text summarization, utilizing two prominent transformer-based models: BART (Lewis et al. 2020) and T5 (Raffel et al. 2020). The experiments were conducted using Python, leveraging the Transformers library (Wolf et al. 2020) for model implementation, Pinecone for vector storage and retrieval, and TensorFlow Hub for embedding generation.

RAG Implementation

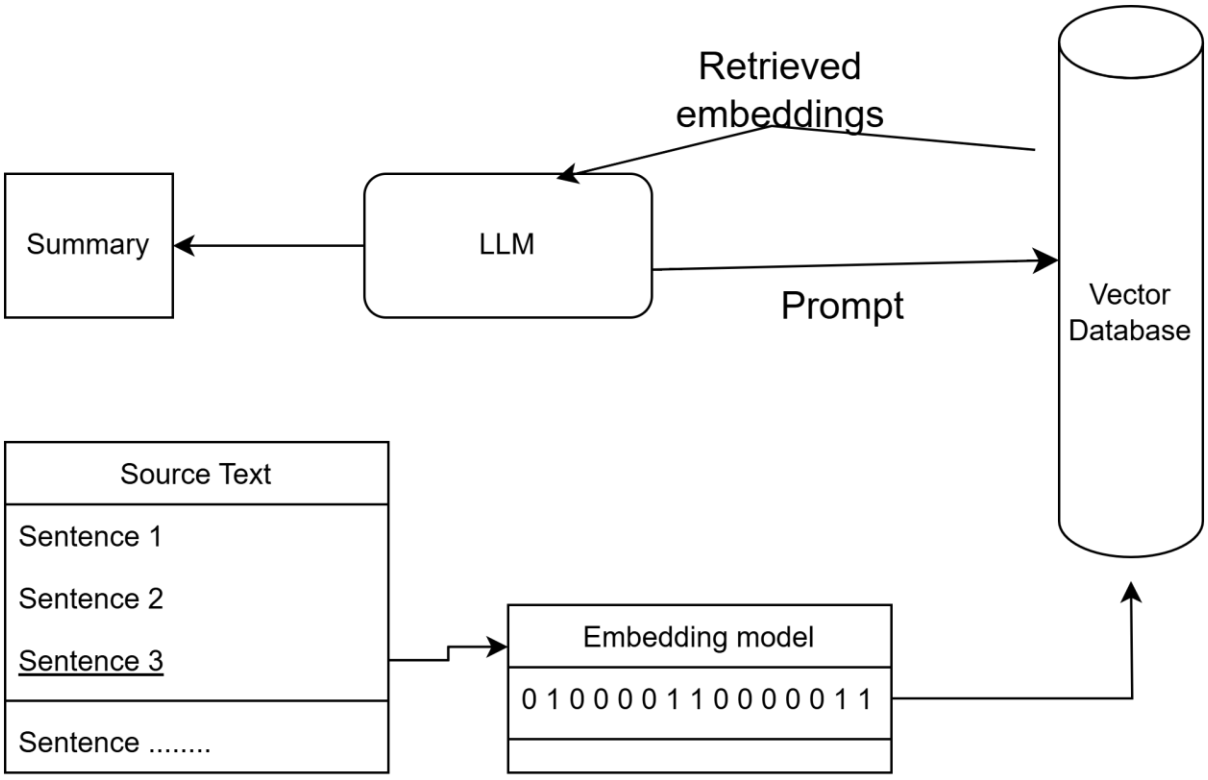

Figure 3: RAG Architecture demonstrating storage of embeddings and retrieval to ground the model in generation

RAG architecture, as shown in figure 3, consists primarily of a retriever and a generation model. The retriever queries the external knowledge base (vector database) to perform a semantic similarity search and retrieves semantically matching text or information, which it then passes to the LLM. The LLM augments its knowledge base with the retrieved information to generate a response matching the received prompt. Similar text or information matching a prompt are ranked and the top_k value determines how strict this ranking process in limiting the number of matches for selection. A fewer number throws up fewer and exact matches for the retriever to select from (Karpukhin et al., 2020).

Summarisation models

Both RAG and non-RAG versions of BART and T5 models were implemented. The non-RAG versions generated summaries directly from the input text, while the RAG versions incorporated retrieved context from the Pinecone database, which was dynamically integrated during the summarization process. This allowed for a comparison of how additional context influences the quality and factual consistency of the generated summaries.

This research followed an earlier work (Liu and Lapata, 2019) to implement a two-step summarization approach was developed. This approach aims to capture comprehensive information in the first step while refining and condensing it in the second step.

The study also experimented with different temperature and top-k values for both BART and T5 models to assess model performance in different environments. Higher temperature values introduced more randomness in the token selection process, potentially leading to more diverse outputs. Adjusting top-k values helps assess the effect of selecting higher-ranked tokens.

To further enhance the RAG implementation, the top-k value used in the retrieval process from the Pinecone database was modified. This adjustment aimed to improve the semantic matching of retrieved sentences, potentially leading to more coherent summaries.

Evaluation metrics

To comprehensively assess the performance of our models, a range of evaluation metrics were employed:

ROUGE scores (ROUGE-1, ROUGE-2, ROUGE-L) measure the overlap between generated summaries and reference summaries (Lin 2004). n-gram is a contiguous sequence of words in a text. While ROUGE-1 measures the overlap of unigram, ROUGE-2 measures the overlap of bigram capturing sequential order. On the other hand, METEOR score evaluates semantic similarity and paraphrase recognition (Banerjee & Lavie 2005), BERTScore captures semantic similarity using contextual embeddings (Zhang et al. 2020), and MoverScore measures the semantic similarity between generated summaries and reference summaries by comparing embeddings of words and phrases, providing a more nuanced evaluation of summary quality (Zhao et al. 2019). In addition, human evaluation was employed to check generated summaries of the first five rows in each and check for hallucinations to compare with automatic metrics and verify consistency with the original text.

Experimental setup

The experiments were conducted in a controlled environment to ensure consistent computational resources across all model variations. The number of training epochs was set to 5 (except for Test 8 set to 15 epochs).

For the RAG implementation, the Pinecone vector database was used to store and retrieve relevant context. The retrieval process was integrated into the model's generation pipeline, allowing for dynamic incorporation of retrieved information during summary generation.

Eight tests using the two models were done. Except for the Test 7 which focused on the import of retrieval factor and Test 8 which tested training epoch factor, all others were done using both RAG and non-RAG implementation. The tables below show the results from each of the tests.

Test 1:

Metric	BART +RAG	T5 +RAG	BART noRAG	T5 noRAG
ROUGE-1	0.653	0.668	0.450	0.578
ROUGE-2	0.579	0.605	0.443	0.542
ROUGE-L	0.612	0.653	0.447	0.563
METEOR	0.509	0.496	0.259	0.385
BERTScore	0.903	0.920	0.902	0.913
MoverScore	0.237	0.190	0.091	0.122

Table 1: Summary evaluation results by models with no injected random facts at temp = 0.7

Test 2:

Metric	BART +RAG	T5 +RAG	BART noRAG	T5 noRAG
ROUGE-1	0.768	0.682	0.527	0.597
ROUGE-2	0.730	0.617	0.517	0.555
ROUGE-L	0.759	0.666	0.527	0.562
METEOR	0.635	0.514	0.340	0.400
BERTScore	0.944	0.923	0.911	0.910
MoverScore	0.246	0.217	0.105	0.128

Table 2: Summary evaluation results by models with no injected random facts at temp = 0.7 & top_k = 75

Test: 3

Metric	BART +RAG	T5 +RAG	BART noRAG	T5 noRAG

ROUGE-1	0.422	0.486	0.229	0.360
ROUGE-2	0.411	0.461	0.219	0.342
ROUGE-L	0.419	0.482	0.229	0.359
METEOR	0.257	0.297	0.114	0.205
BERTScore	0.894	0.904	0.867	0.886
MoverScore	0.091	0.099	0.069	0.084

Table 3: 2-step summarisation evaluation results by models with no injected random facts at temp = 0.7 & top_k = 75

Test:4

Metric	BART +RAG	T5 +RAG	BART noRAG	T5 noRAG
ROUGE-1	0.299	0.703	0.505	0.577
ROUGE-2	0.038	0.646	0.451	0.516
ROUGE-L	0.179	0.690	0.483	0.548
METEOR	0.177	0.554	0.329	0.395
BERTScore	0.733	0.921	0.882	0.904
MoverScore	0.217	0.267	0.119	0.128

Table 4: Summary evaluation results by models with injected random facts at temp = 0.7

Test 5:

Metric	BART +RAG	T5 +RAG	BART noRAG	T5 noRAG
ROUGE-1	0.674	0.681	0.442	0.574
ROUGE-2	0.613	0.612	0.407	0.509
ROUGE-L	0.641	0.653	0.419	0.544
METEOR	0.541	0.527	0.263	0.388
BERTScore	0.914	0.915	0.886	0.900
MoverScore	0.265	0.258	0.095	0.130

Table 5: Summary evaluation results by models with injected random facts at high temp= 1. 5 & top_k = 75

Test 6:

Metric	BART +RAG	T5 +RAG	BART noRAG	T5 noRAG
ROUGE-1	0.397	0.481	0.239	0.347
ROUGE-2	0.369	0.454	0.211	0.308

ROUGE-L	0.378	0.472	0.224	0.333
METEOR	0.232	0.296	0.120	0.195
BERTScore	0.878	0.898	0.855	0.873
MoverScore	0.092	0.103	0.071	0.083

Table 6: 2-step summarisation evaluation results by models with injected random facts at temp = 0.7

Test 7:

Test 7		
Metric	**BART +RAG**	**T5 +RAG**
ROUGE-1	0.472	0.499
ROUGE-2	0.458	0.467
ROUGE-L	0.468	0.492
METEOR	0.289	0.311
BERTScore	0.904	0.903
MoverScore	0.094	0.104

Table 7: Summary evaluation results by models with no injected random facts with top_k retrieval value set to 3

Test 8		
Metric	**BART +RAG**	**T5 +RAG**
ROUGE-1	0.746	0.685
ROUGE-2	0.708	0.627
ROUGE-L	0.731	0.678
METEOR	0.612	0.524
BERTScore	0.938	0.927
MoverScore	0.232	0.247

Table 8: Summary evaluation results by improvement of Test 2 by increasing epoch to 15

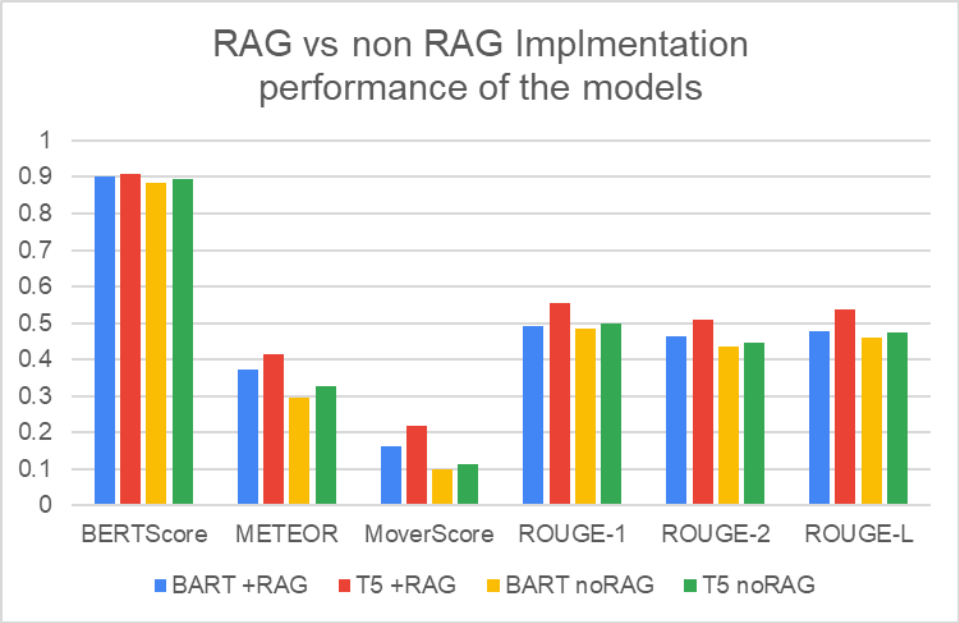

Figure 4: Summary performance of models with RAG vs no RAG

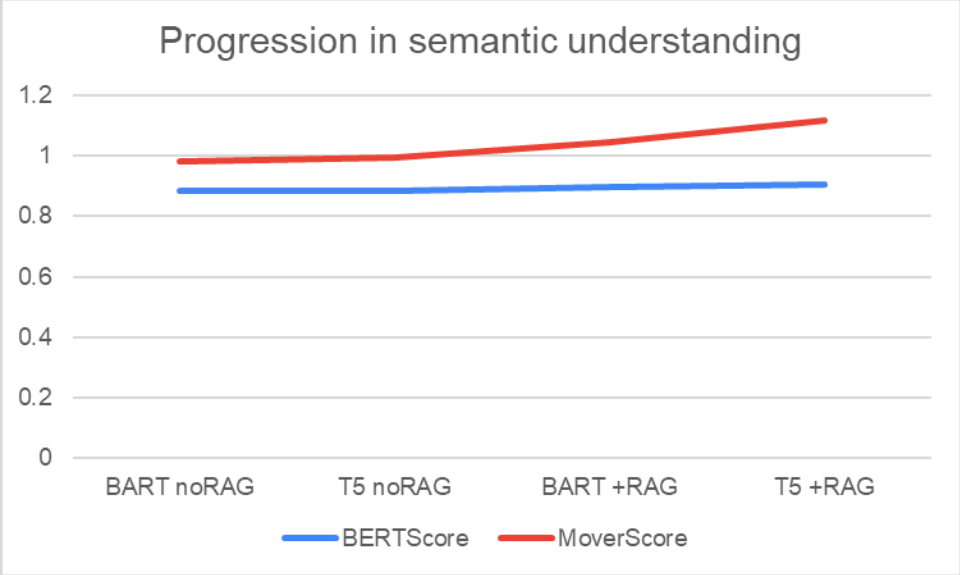

Figure 5: Summary performance of the models showing semantic understanding

Hypothesis

The next step is to discuss our hypothesis in the light of the results above:

- RAG implementation improves the quality of summarization.
- The 2-step summarization technique results in better summaries.
- MoverScore is a better measure of semantic similarity than other metric in summaries.

Summaries with RAG consistently outperformed summaries 6 without RAG as can be seen from Table 1 to Table and more clearly in Figure 4, especially when compared within models. Comparing RAG against non-RAG implementation, BART showed 21% improvement with the introduction of RAG whereas T5 recorded a 17.3% increase with RAG implementation. Overall, T5 consistently outperformed BART with or without. However, BART with RAG in Table 2 is the best model and the model with the best MoverScore result.

Results from Table 3 against Table 1 on one side, and Table 6 against Table 4 do not support the hypotheses that a 2-step summarization technique brings about improvement in summary quality. All results from Table 1 without injected text and Table 4 with injected text that performed singe step summarisation were better than their counterparts in Tables 3 and 6 where 2-step summarisation was adopted.

While both RAG and non-RAG implementations compare favourably in BERTScore, RAG implementations improve clearly above non-RAG implementations in MoverScore. As seen in Figure 5.

Discussions

Tests in Tables 1, 2, 3, 7 and 8 were conducted without injected facts (noise). Except for Test 2 and 8, conducted with models set at a temperature value of 1.5 and top_k value of 75, all other tests were conducted with a temperature value of 0.7 and top_k value of 50. BART with RAG in Test 2 produced the best model. Models in Test 2 performed better than models in Test 1,3 and even in Test 7 when top_k retrieval value was tweaked to 3 and to 7. Test 8 was performed to improve Test 2 and test factors affecting retrieval and training epoch increased by 150%, T5 improved marginally by 1.3% and BART decreased by 2.9%. This helps us to conclude that a combination of the retrieval process and parameter settings of the model is necessary to improve summary generation.

Test in Tables 4,5,and 6 were conducted with injected texts. T5 with RAG consistently handled noise better than BART with RAG showing 32.7% improvement. Test 5 at a temperature of 1.5 and top_k value of 75 performed better than Tests 4 and 6 for RAG implementation whereas the reverse was the case for non-RAG implementation. T5 with RAG produced its best model in Test 5when compared with Tests 1 – 7 demonstrating its robustness to handle noise better than other model.

Surprisingly, manual check of generated summaries reveals that higher evaluation scores does not translate to low hallucination in generated summaries. This work argues that

tools like ACUEVAL (Wan et al., 2024) that identifies hallucinations and their locations in a generated summary and takes an iterative approach to resolution of hallucination could be a better approach to hallucination evaluation and mitigation.

This paper agrees with previous works of Lewis et al (2021), Deepthi and Sowjanya (2021) and Karpukhin et al. (2020) on the merits of embeddings in NLP downstream tasks. To the best of our knowledge, this work is the first to propose the use of the source text for summarisation as the text for embedding from which the retrieval process updates the summarisation model. This work extends the research by Karpukhin et al. (2020) in Question Answering, which outperformed traditional BM25 in retrieval accuracy by 9% to 19%, to text summarisation. This work argues that combining an improved retrieval process with model parameter adjustments can significantly enhance summary generation. Our findings demonstrate that this approach yields at least a 17.3% improvement in quality of generated summaries.

Conclusion

This work demonstrates that the use of Retrieval-Augmented Generation (RAG) is an effective strategy for mitigating hallucinations in abstractive text summarization. RAG's ability to tap into an external knowledge base significantly enhances the accuracy and factual grounding of large language models (LLMs), leading to measurable improvements in summary quality. Quantitatively, BART with RAG showed a 21% improvement over its non-RAG counterpart, while T5 with RAG achieved a 17.3% improvement, underscoring the effectiveness of RAG across different models.

However, the persistence of hallucinations, even with RAG, suggests that RAG may need to be supplemented with additional approaches in certain contexts. A combination of retrieval process management and model parameter setting is needed for improved summary output by LLMs. Tools like ACUEVAL can serve as better evaluation metric as it not only provides explanation but also helps to mitigate hallucination. This study highlights the need for future research to further refine the retrieval process, apply these methods to larger and more diverse datasets, and explore the potential of these techniques in long-text summarization.