SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Analyzing and Improving Post-hoc Approaches for the Detection and Correction of Hallucinations in Long-form Text Generation

Ihsan Soydemir

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Analyzing and Improving Post-hoc Approaches for the Detection and Correction of Hallucinations in Long-form Text Generation

Analysieren von und Verbesserung von Post-hoc-Ansätzen zur Erkennung und Korrektur von Halluzinationen bei der Generierung von Langformtexten

Author:	Ihsan Soydemir
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Juraj Vladika, MSc
Submission Date:	08.10.2024

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Location, Submission Date

Author

AI Assistant Usage Disclosure

Introduction

Performing work or conducting research at the Chair of Software Engineering for Business Information Systems (sebis) at TUM often entails dynamic and multi-faceted tasks. At sebis, we promote the responsible use of *AI Assistants* in the effective and efficient completion of such work. However, in the spirit of ethical and transparent research, we require all student researchers working with sebis to disclose their usage of such assistants.

For examples of correct and incorrect AI Assistant usage, please refer to the original, unabridged version of this form, located at this link.

Use of *AI Assistants* for Research Purposes

I have used AI Assistant(s) for the purposes of my research as part of this thesis.

X Yes ◯ No ◯

Explanation: I used Grammarly for corrections and Large Language Models (LLMs) to improve the structure and flow of my writing.

I confirm in signing below, that I have reported all usage of AI Assistants for my research, and that the report is truthful and complete.

Location, Date

Author

Acknowledgments

I extend my heartfelt gratitude to Prof. Florian Matthes for supervising my thesis. I also sincerely thank Juraj Vladika, MSc, for his guidance and support from start to finish. Without his help, this thesis would not have come this far or reached the finish line.

I am also deeply grateful to my parents for all their sacrifices and constant support. Lastly, to my soulmate, thank you for your love, patience, and always being there.

Abstract

Detecting and correcting hallucinations in text generated by large language models is still a big challenge, even for the most advanced systems. Many approaches have been suggested to solve this issue, but none have fully addressed all the difficulties. This thesis focuses on post-hoc methods on long-form text generation. Post-hoc stands for identifying and fixing hallucinations after the text is generated.

While current post-hoc techniques show some success with short, single-sentence claims, they often struggle when applied to longer-form content. This research aims to systematically analyze existing methods for mitigating hallucinations in long-form text. This work will also introduce a taxonomy to categorize different types of hallucinations. The strengths and limitations of current approaches will be evaluated, and enhancements will be proposed to improve their effectiveness in handling long-form generation.

Kurzfassung

Das Erkennen und Korrigieren von Halluzinationen in Texten, die von großen Sprachmodellen generiert wurden, ist selbst für die fortschrittlichsten Systeme immer noch eine große Herausforderung. Viele Ansätze wurden vorgeschlagen, um dieses Problem zu lösen, aber keiner hat alle Schwierigkeiten vollständig bewältigt. Diese Arbeit konzentriert sich auf Post-hoc-Methoden für die Generierung von Langformtexten. Post-hoc bedeutet, dass Halluzinationen nach der Texterstellung identifiziert und korrigiert werden.

Während aktuelle Post-hoc-Techniken bei kurzen, einsilbigen Behauptungen einen gewissen Erfolg aufweisen, haben sie bei der Anwendung auf längere Inhalte oft Schwierigkeiten. Ziel dieser Forschungsarbeit ist die systematische Analyse bestehender Methoden zur Entschärfung von Halluzinationen in langen Texten. Diese Arbeit wird auch eine Taxonomie zur Kategorisierung verschiedener Arten von Halluzinationen einführen. Die Stärken und Grenzen aktueller Ansätze werden evaluiert, und es werden Verbesserungen vorgeschlagen, um ihre Effektivität bei der Erstellung von Langformtexten zu verbessern.

Contents

1 Introduction

In first chapter, the motivation behind this thesis will be explained along with research questions that are addressed through the course of this study.

1.1 Motivation

Large Language Models (LLMs) are introducing a paradigm-shift in the field of NLP [1]. LLMs are general-purpose and have started to replace dedicated models tailored for NLP sub-tasks [2]. Ongoing research and academia trends demonstrate that LLMs are gaining popularity in solving specific problems thanks to the domain-specific knowledge obtained via pre-training phase. Text classification [3], [4], [5], machine translation [6], [7], summarization [8], [9] are only some subset of downstream NLP tasks LLMs can achieve without further fine-tuning, unlike earlier practices.

However, despite their capabilities and increasing popularity [10], LLMs, due to their nature, are considered prone to hallucinations [11]. Hallucination is when LLM generates inaccurate or fabricated text. This misleading behaviour of LLMs is often difficult to detect because generated text is often convincing, detailed, realistic, and plausible.

In this age of misinformation [12], it is extremely important to distinguish between fact and misinformation. While there is already reasonable amount of fabricated information produced and being produced daily, generative AI models, especially LLMs, do not also guarantee any sort of truthfulness. At this point, the need to develop systems to eliminate this weakness of LLMs has become apparent, as people are constantly inundated with false information from various sources.

Recent studies indicate that LLM-generated text will always hallucinate [13]. Therefore, it is undeniable that the outputs produced from these text generators must be filtered for accuracy using some advanced techniques [14]. In this study, a subset of these techniques, post-hoc methods, will be analyzed. These frameworks focus on correcting hallucinations after text is generated. Refining the text after generation phase is called post-hoc correction. While post-hoc factuality improvements have been widely explored for short-text, detection and correction of hallucinations in long-form text is the main focus of this study.

1.2 Research Questions

This research aims to analyze and improve post-hoc approaches for detecting and correcting hallucinations in long-form text generation. Following three research questions will be addressed:

RQ1 What is an appropriate taxonomy for categorizing hallucinations in large language models (LLMs), such as numeric and semantic hallucinations?

RQ2 How can the Retrieval Augmented Generation (RAG) technique be effectively applied to handle long contexts without compromising performance or efficiency?

RQ3 How can editing and faithfulness be optimally balanced when refining generated text?

2 Background

This chapter focuses on the building blocks of this thesis, and explains the following terms: hallucination, factuality and faithfulness in Large Language Models (LLMs), fact-checking, and fact-checking systems.

2.1 Hallucinations

Hallucination is a term often associated with its psychological context [15]. A simple explanation of this concept is an incorrect perception of outside world from an individual [16].

In AI context, the term hallucination first appeared in computer vision [17] in early 2000s. Hallucination was once considered helpful for several sub-tasks in computer vision such as super-resolution, inpainting, etc [18], since the nature of these tasks depends on generation or modification of non-existing pixels. Recent CV work also mentions hallucination [19], which holds a negative meaning. For example, in [20] hallucination refers to incorrect captions for images - i.e. mentioning objects that do not exist within the provided image.

Natural Language Generation (NLG), is one of the main areas in which the term hallucination is relevant. Within NLP context, NLG refers to generation of text, recently demonstrating a wide range of success with Transformer-based [21] language models [22], [23], [24]. Some example tasks are: summarization, text completion, machine translation, etc. Despite these advancements, recent research shows that NLG models, more specifically LLMs, tend to generate text that is hallucinated [25], [26]. Similar to psychological and computer vision context, hallucinations in NLG also refer to faithfulness and factuality issues in [27] generated text.

2.2 Factuality and Faithfulness in LLMs

With emergences in Natural Language Generation (NLG) models, researchers and industry have started exploring applications of Large Language Models (LLMs). Introduction of a chat interface that allows simple interaction and accessibility of these models through a single API also increased their popularity. This ease of use also allowed exploration of their use cases for many different areas. Applications of LLMs exploded, but there is a side effect that comes with that: LLMs provide plausible text without any form of guaranteed factuality.

Researchers have also started working on hallucinations in text generations. It is a challenging problem since LLMs are autoregressive models and refuse to say “I don’t know”.

Many aspects of hallucinations are still an ongoing research question — from categorization to mitigation.

On classification side of hallucinations, a common way is proposed as faithfulness and factuality in the first level of classification [28]. There are also recent works that explore causes of hallucinations, that focus on following categorizations: data, training, and inference. [11].

There are also a lot of ongoing efforts to detect hallucinations. A common approach is detecting them through retrieving external facts, FActScore [29], FACTKG [30], and FacTool [31] are some examples. Benchmarks such as TruthfulQA [32] and RealTimeQA [33] have been proposed to detect issues with faithfulness and factuality.

2.3 Fact Checking Systems

Research focus is not only limited to the classification, causes, detection, and benchmarking of hallucinations but also mitigation of them. This is when fact-checking systems, also known as fact-checkers come into play.

Fact-checking means checking if generated text or a claim is true based on evidence. Automated methods have been developed to make this process easier. In this study, the main focus will be on post-hoc LLM-based fact-checkers that are capable of re-writing given input text by replacing hallucinated parts. Post-hoc correction approach is powerful, works directly, applicable to both closed and open models.

Fact-checking in the scientific context and news ecosystem is quite important since hallucinations pose significant problems, particularly when text generation models are used in critical domains, such as news, or healthcare [34]. This is even more challenging for long-form text.

In general, LLM-based fact-checkers that apply post-hoc techniques employ a systematic pipeline. These pipelines usually start with the atomic sentence creation step. This step divides a generation into a set of atomic facts. Question generation phase follows, and gives a list of questions to validate. Each atomic fact is then validated by passing those question(s) to some knowledge source. The final step is the generation of the corrected claim.

3 Related Work

This chapter will explain LLM-based fact-checking methods and approaches, focusing primarily on COVE and RARR. Improvements for these fact-checking frameworks will be proposed in the later chapters by referring to various experiment results.

3.1 CoVE

Chain-of-verification, or in short COVE [35], introduced a framework to improve factuality of language model generated text. It is developed by Meta AI team. They mention that if LLMs generate text to given queries without any planning it is more likely to result in hallucinated output. This approach proposed multiple steps before giving the final response, in order to reduce hallucinations. Pipeline is shown below in Figure 3.1.

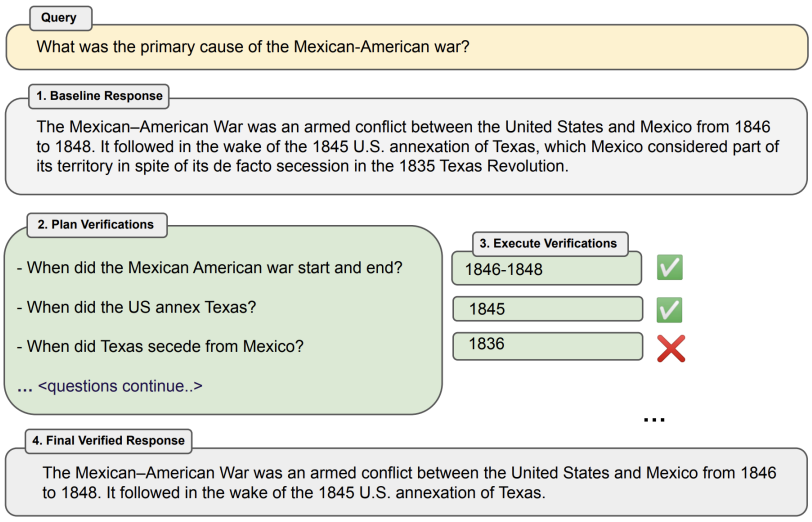


Figure 3.1: COVE Pipeline [35]

These steps include:

1. Baseline Response Generation: Given a question, COVE generates answers. This is a very straight-forward step in the pipeline, without any adjustments.
2. Verification Planning: After baseline response is generated, COVE asks LLM to extract questions from this baseline response. This step is quite critical because in case of irrelevant/incorrect question generation, further steps might fail to recover.

3. Execution of Verifications: Following the planning of verification, now LLM is asked to answer each of these questions to identify any potential hallucinations.

4. Generation of Refined Text: Creates a new, refined version using the previous steps. Here, verification question-answer pair list is also given as context.

COVE mentions that all steps, except executions for step 3, can use a single prompt. However, they state that the execution of verification prompting can follow four different types. These four types are: Joint, 2-Step, Factored, and Factor+Revise. Each of these approaches varies in that they can involve a single prompt, two prompts, or independent prompts for each question.

COVE relies only on internal knowledge. Internal knowledge in LLMs is also referred to as "intrinsics" or "self-knowledge.". It is possible to extend the proposed framework to use search engines, the authors did not explore this or any other tool use.

3.2 RARR

Researching and Revising What Language Models Say, Using Language Models, RARR [36], is another fact-checker, developed by a joint work of CMU, Google, and UC Irvine. This work is similar to COVE because it follows four steps, just like COVE does, and each step serves a quite similar purpose as the steps in COVE. RARR fact-checking relies on Research & Revision steps. Another significant difference is that RARR explores and uses external tools. Pipeline is included in below Figure 3.2:

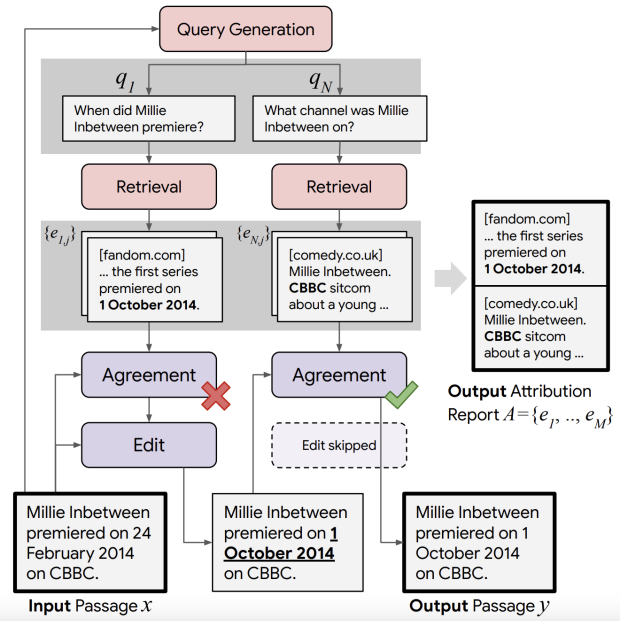


Figure 3.2: RARR Pipeline [36]

Research stage includes the following steps:

1. Query generation: RARR generates questions using CQGen (comprehensive question generation). The main goal is to cover all details of the given passage.

2. Evidence retrieval: Google Search Engine is used to answer the generated questions. Five web pages are retrieved to find a relevant answer. Navigating through each website and retrieving the full text is the next step. Finally, a 4-sentence sliding window runs through the sentences. These evidence passages, or chunks, then receive a relevance score based on the query, and the top $J = 1$ evidence is kept for each query.

Revision stage includes the following steps:

1. Agreement model: RARR uses a prompt-based logic implementation, i.e., question-guided agreement, as also mentioned in [37].

2. Edit model: RARR uses an agreement gate to check if a given claim aligns with provided evidence. If it aligns, it uses an edit model to revise claim. Revision is only applied if the change (text distance) is small.

RARR authors avoid using the term "fact-checking". Instead, they use "attribution" because even if a claim is verified using a source, it does not guarantee that the source is "correct". They also include a final attribution report, where relevant evidence snippets are extracted.

RARR accepts claims as input; however, questions do not result in any claim generation. Therefore, the RARR pipeline does not directly support asking a question as COVE does. Questions need to be answered first, and in this thesis, LLMs are used to generate claims to allow RARR to fact-check.

4 Methodology

This chapter introduces used datasets, evaluation methods, and taxonomy proposed to answer research questions aforementioned.

4.1 Datasets

In this section, datasets used in the experiments will be introduced. First two, SummEdits and ExpertQA contain gold answers, whereas LongFact does not. These datasets serve as a benchmark for evaluating model and methodology performances.

4.1.1 SummEdits

SummEdits is a benchmark to test how well large language models (LLMs) can detect factual inconsistencies in summaries. It focuses on determining whether edits to summaries keep the content consistent or introduce errors. The benchmark works across 10 domains, but this work focuses on the news subcategory.

The domains include bill summaries, e-commerce summaries, news articles, podcast transcripts, Q&A summaries, sales call transcripts, sales email summaries, dialogue summaries, scientific TL; DRs, and Shakespearean texts.

Hallucinated Summary	Label	Original Summary	Edit Types	Source
National air traffic management agency Airservices Australia says it has shifted from **delivery** to **prototyping** phase of an effort to deploy an airspace management tool for small, uncrewed aircraft such as **military** unmanned air vehicles.	0	National air traffic management agency Airservices Australia says it has shifted from **prototyping** to the **delivery** phase of an effort to deploy an airspace management tool for small, uncrewed aircraft such as **commercial** unmanned air vehicles.	antonym_swap, entity_modification	Australia's air traffic management agency will deploy an airspace management system for small, uncrewed aircraft systems (UAS) in 2025. Airspace Australia on 27 February said it has completed prototyping on its Flight Information Management System (FIMS) for UAS and is now focused on delivering a product in 2025. ...

Table 4.1: SummEdits Dataset Sample - News Category. Bold text shows edits.

As shown in Table 4.1, this dataset applies various edit types to the original summary to create a hallucinated summary. In this thesis, SummEdits dataset is widely used in experiments — it allows to analyze and evaluate fact-checkers using gold-standard answers.

4.1.2 ExpertQA

ExpertQA is a question-answering dataset. In this dataset, questions are answered by humans who are domain experts. This dataset contains newline-separated JSON entries (jsonl files) with specific fields relevant to the questions and answers. Each question is written by an expert. Answers are generated using up to date LLM models at the time work has been published i.e. GPT-4, Bing Chat, etc.

Dataset is given in JSONL format. For simplicity, only a couple of key fields relevant to the understanding of the dataset are highlighted below:

- *question*: Question written by an expert – from a domain (e.g. physics, law, etc).
- *answers*: Dictionary that gives answers. Maps names of different models to *Answer*. Keys (models) are `gpt4`, `bing_chat`. They also include fine-tuned GPT-4 versions (i.e. `rr_sphere_gpt4`). Each answer object also includes the following:
  - *answer_string*: Original answer (Model generated)
  - *attribution*: List of URLs that provide evidence for the answer
  - *claims*: List of claims made within the answer. They also give more interpretability by adding supporting evidence.

- *revised answer string*: Revised answer that has been refined by an expert annotator. Annotators improve clarity and factuality since they are experts in this subject.

Below is an example of a question and its corresponding answers, with many keys ignored for the sake of simplicity:


```
{
  "question": "Imagine that EU no longer exists. What influence will it have
     for Poland economics?",
  "answers": {
    "post_hoc_sphere_gpt4": {
      "answer_string": "If the European Union (EU) were to no longer exist, the
           Polish economy would likely experience significant consequences in
          key areas [...] Consequently, Polish exports might decline and
          businesses could face increased costs [...] might struggle to maintain
           or increase these investment inflows [...] economic growth and job
          creation...",
      "revised_answer_string": "If the European Union (EU) were to no longer
          exist, the Polish economy would likely experience significant
          consequences in key areas [...] Firstly, trade would be impacted as
          the EU accounts for about 81% of Polish exports and 58% of its imports
           [...] Secondly,..."
    }
  }
}
```


4.1.3 LongFact

LongFact is a long-form dataset from long-form-factuality [38]. It is similar to the MMLU-style datasets and contains questions that require detailed answers. These questions can be answered by large language models. Authors used GPT-4-0613 model from OpenAI to answer questions generated. However, the generated answers were not publicly available in the GitHub repository at the time this paper was written. Still, generation of these answers using an LLM is straightforward and not a significant challenge. It is important to note that, there is no gold answer provided in this dataset, unlike ExpertQA or SummEdits.

The questions are divided into two categories: Concepts and Objects. The given dataset covers 38 topics across a wide range of fields, including physics, biology, machine learning, international law, architecture, accounting, sociology, and virology. It is also possible to enhance this dataset with custom topics.

"Object" questions are easier to address. As they focus on specific, tangible entities, such as scientific instruments or experiments, and require straightforward factual descriptions.For example:

- "Can you tell me about TRIUMF, Canada's national particle accelerator facility?"

- "What can you tell me about the LIGO Scientific Collaboration (LSC)?"

In contrast, "Concept" questions deal with abstract ideas like nuclear fusion or quantum entanglement. It requires a more in-depth explanation of theoretical principles, processes, or implications. Concept questions demand deeper exploration, while object questions are based on well-documented knowledge. For instance:

- "How does the principle of quantum entanglement challenge the traditional laws of cause and effect?"
- "Could you expound on the principles of Quantum Field Theory, including its foundations in quantum mechanics and special relativity, its description of forces via exchange particles, key phenomena such as spontaneous symmetry breaking and virtual particles, and its implications for our understanding of the fundamental forces and particles of the universe?"

In essence, the task of LongFact is to generate comprehensive, long, and detailed answers to questions about asked concepts or objects.

4.2 Evaluation

SummEdits and ExpertQA datasets include gold summaries. They allow the utilization of reference-based metrics. Therefore, the following metrics - except FActScore - have been used by providing LLM-generated text along with the respective ground truth. The goal is to evaluate factuality, relevancy, and quality of the generated text.

4.2.1 BERT Score

BERTScore [39] is a text generation metric that uses contextual embeddings. It measures semantic similarity, but it's not great for detecting hallucinations. It captures context and shared words, even when sentences aren't factually related. Hallucinated content can still score high if it sounds plausible or uses similar phrasing. BERTScore doesn't focus on factual accuracy, so it is generally not able to pay attention to false information. For hallucination detection, more advanced techniques should also be explored, but this metric is helpful to understand the relevancy.

4.2.2 Semantic Similarity

Semantic similarity measures how closely two texts align in meaning. In this work, SimCSE model [40] is used to embed both the reference and the generated summary, then compute their cosine similarity.

In hallucinated text correction experiments, both BERTScore and semantic similarity scores were found to be quite high. Yes, these metrics effectively indicate the overall relevance of the generated content. But they are not particularly useful for identifying hallucinations. They

provide an idea of how related the texts are but do not guarantee factual accuracy. Therefore, they should be complemented with other evaluation methods.

4.2.3 Text Distance Metrics

Edit Distance

A metric on Σ^* is defined by the edit distance, also known as the Levenshtein distance [41]. The edit distance between two strings s and t is the minimum number of edit operations required to transform s into t (or vice versa).

Following are edit operations:

1. Substituting one character with another.
2. Deleting a character.
3. Inserting a character.

In short, edit distance measures how distinct two strings are by calculating the smallest number of operations necessary to convert one string into the other.

Normalized Edit Distance

Normalized edit distance [42], by Li and Liu, introduces a new normalized edit distance for comparing two strings. It's based on their lengths and Levenshtein Distance. This method is a true metric, meaning it satisfies the triangle inequality, unlike some other normalized edit distances.

Word Overlap (ROUGE-1)

Word overlap, also known as ROUGE-1 [43], measures text similarity by counting the overlap of individual words (unigrams) between the reference and the candidate text. A higher overlap indicates greater similarity.

Bigram Distance (ROUGE-2)

Bigram distance, referred to as ROUGE-2 [43], evaluates text similarity based on the overlap of consecutive word pairs (bigrams). It helps to provide a more nuanced comparison by considering word order and context.

Text distance-based metrics complemented the following two main metrics, NLI and G-Eval.

4.2.4 NLI

NLI Score is a metric based on the principle of natural language inference. It is also known as textual entailment [44]. It interprets the reference response as the hypothesis and the generated answer as the premise. The underlying idea is that a good answer should logically

follow the reference answer. This NLI-based method has been applied to evaluate the quality of summaries ([45]; [46]; [47]). More specifically, DeBERTa-v3 model [48] is utilized. The fine-tuned version of this model on various NLI datasets is employed, as it performs well with long text [49]. The model predicts scores for each class (entailment, neutral, and contradiction) between 0 and 1. Scores sum up to 1.

4.2.5 G-Eval

G-Eval [50] is a framework based on LLM prompting with chain-of-thoughts to evaluate the quality of generated texts in a form-filling paradigm. It is one of the most popular "LLM-as-judge" metrics [51], which assesses the LLM output using another LLM with finely crafted prompts and takes the numerical output as the final score. The original study found G-Eval to be highly correlated with human judgment.

The implementation of G-Eval is primarily used from the repository deepeval, developed by Confident AI [52]. In this thesis, 3 G-Eval metrics are defined (Higher is better for all the following):

- *General Evaluation*: Evaluates overall quality by assessing coherency, accuracy, factual inaccuracy, and correct language.
- *Hallucination*: Penalizes heavily if actual output contains hallucinated information not present in input.
- *Relevency*: Boosts score if actual output is relevant to the given input, penalizes off-topic responses and irrelevant information.

These metrics are then detailed during the implementation phase. Each metric consists of a name, criteria, evaluation steps, parameters, and the LLM model (e.g `gpt-3.5-turbo-0125`) to be used for evaluation.

4.2.6 FactScore

Fine-grained atomic evaluation of factual precision in long-form text generation, or FactScore [29], is also a framework that allows evaluating LM generated long-text. The main idea consists of 2 key concepts:

- *Creating atomic facts out of sentences*: Even one sentence can have multiple facts to fact-check against.
- *Choosing source of knowledge*: Truthfulness or factuality is a subjective concept. Therefore, FactScore consider if a claim is true or false by selected knowledge source.

Definition. Suppose $\mathcal{M}$ gives a response $y = \mathcal{M}_x$ for a prompt $x \in \mathcal{X}$, and A_y is the list of atomic facts in y. The *FactScore* of $\mathcal{M}$ is defined as:

$$f(y) = \frac{1}{|A_y|} \sum_{a \in A_y} \mathbb{I}[a \text{ is supported by } C],$$

where $\mathcal{M}$ is a LM, $\mathcal{X}$ is the set of prompts, and C is the knowledge source.

$$\text{FactScore}(\mathcal{M}) = \mathbb{E}_{x \in \mathcal{X}}[f(\mathcal{M}_x) \mid \mathcal{M}_x \text{responds}].$$

Here, $\mathcal{M}_x$ responds means that $\mathcal{M}$ provides an answer to prompt x. This definition assumes:

1. It's clear whether an atomic fact is supported by C.
2. All atomic facts in A_y have equal importance ([53]).
3. Information in C is consistent, with no internal conflicts.

The final FactScore represents the percentage of atomic facts supported by an external source. Even short texts can create a large number of atomic facts since the coverage is high and the granularity is low. FactScore has been only used to evaluate the third dataset (LongFact) without using any references.

Adjustments

In this thesis, FactScore official implementation [54] from GitHub is used throughout the experiments, with some modifications, which include:

1. ***Adopting a Larger Wikipedia Dump***: For experiments on LongFact, default English Wikipedia Dump (2023/04/01) some pages were found to be missing. Therefore, this dump has been replaced by an older version, a larger and more comprehensive dump (2017/08/20). It performed better, Similar to the default dump, it contains the article topic, section title, and text of the article.
2. ***Finding Missing Topics***: During initial experiments with FActScore pipeline, generated text and the knowledge source were not enough to get a final score. Main issue was the lack of the topic parameter. In Wikipedia context, topic is equal to the article title. Following simple prompt is integrated into the existing pipeline, to get missing article title and run without errors:

```
"""
Please give me the respective Wikipedia article title for following text:
{llm_generated_long_answer}.

Only reply with the title. Do not make any other comments or explanations.
Do an online search to find the exact title.
"""
```

After providing a comprehensive Wikipedia dump and providing missing parameters (article titles), LongFact FactScore has been generated.

4.3 Taxonomy

This section introduces a new taxonomy of hallucination types. Main motivation is to understand how hallucinations are distributed and evaluate which type is easier to detect and mitigate.

Proposed Taxonomy

The proposed taxonomy for hallucination types is given in the Figure 4.1 below:

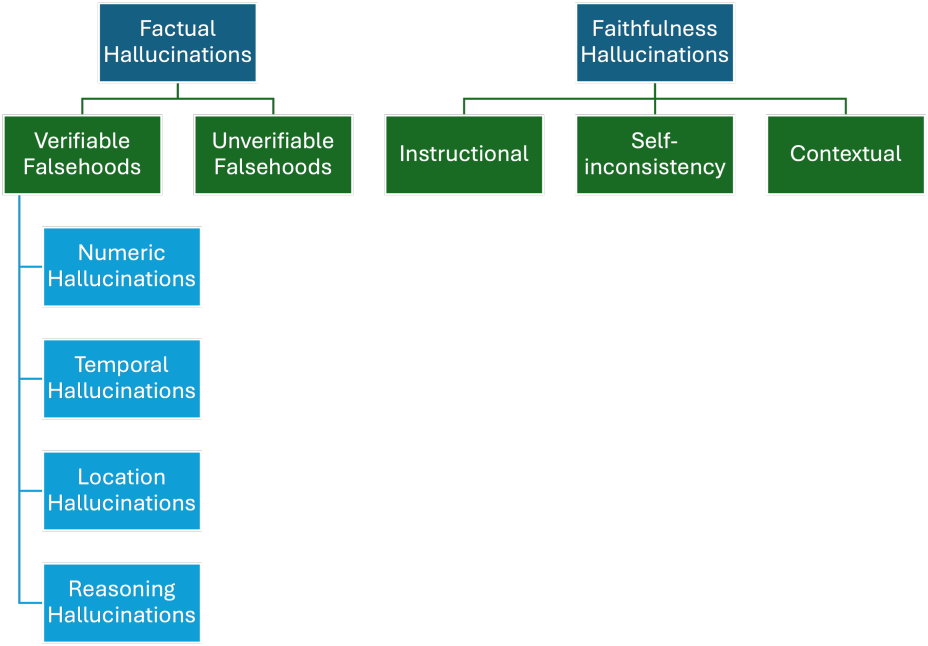


Figure 4.1: Proposed taxonomy. Inspired by [14], [11]

The first level consists of two types: the first is factual hallucinations. They are simply factually incorrect claims/statements). Factual hallucinations consist of two sub-categories:

1. Verifiable Falsehoods: Claims with falsifiability or refutability, that can be proven to be wrong assuming there is evidence

2. Unverifiable Falsehoods: Claims that are either imaginary or can not be proven to be wrong

Second, responses that deviate from a given context, message history, or instructions are called faithfulness hallucinations. The first category of faithfulness hallucinations is instructional, where LLM response does not follow the prompts. Self-inconsistency is caused by contradicting responses either within the same message or a long chat history. Lastly, contextual hallucinations are referred to as misrepresentations of context facts.

This is followed by subcategories for verifiable falsehoods:

- *Numeric Hallucinations*: Incorrect numerical information.
- *Temporal Hallucinations*: Incorrect date or time information.
- *Location Hallucinations*: Incorrect geographical, place-related information.
- *Reasoning Hallucinations*: Faulty or does not follow logical reasoning.

More detailed overview of the hallucination types and examples are given in Table 5.6 below:

Type	Category	Subcategory	Example	Explanation
Factual	1.1 Verifiable Falsehoods	1.1.1 Numeric Hallucinations	Current population of Munich is **less than 1 million**.	Actual population is over 1.5 million.
		1.1.2 Temporal Hallucinations	Galatasaray won the UEFA Cup in **1999**.	They actually won in 2000.
		1.1.3 Location Hallucinations	Eiffel Tower is located in **Berlin**.	It is actually in France.
		1.1.4 Reasoning Hallucinations	If it rains, the ground gets wet; the ground is wet, so it **must have rained**.	There can also be another cause such as a spill.
	1.2 Unverifiable Falsehoods		Joe woke up at **7 am** today.	Simply can not be proven wrong, and does not have to be an imaginary concept.
Faithfulness	2.1 Instructional		User: "How can I make dairy-free **pasta** sauce?" LLM: "Use mozzarella to make a dairy-free **pizza** sauce."	It is not following the instructions and gives the recipe for the pizza instead of pasta.
	2.2 Self-inconsistency		Message 1: "Hamburger is a **healthy** food." Message 2: "Fast-food can be **harmful to health**."	LLM does not follow its own reasoning in multi-turn conversation — hamburger is a fast-food can not be healthy if it is harmful.
	2.3 Contextual		Context: "Tomatoes are widely used **fruits** in Mediterranean countries." LLM: Tomatoes are commonly used **vegetables** in the Mediterranean region.	LLM contradicts the context and uses vegetables instead of fruits. Real-world or common knowledge is not relevant here — context is what matters.

Table 4.2: Taxonomy of Hallucinations with Examples and Explanations

This taxonomy is proposed to better understand the distribution of hallucinations. It is quite important to have some deeper insight on hallucinations while still keeping a high-level perspective. For datasets used, especially for SummEdits, this taxonomy is well-suited since it covers edit or perturbation types applied to the original summary to create a hallucination version of it. Under results section 5.3. (Distribution of Hallucination Types) more insights on hallucination types will be provided in the light of experiments.

5 Results

In this chapter, primarily, the analysis of existing fact-checkers with different setups will be discovered. Improvements on fact-checkers will be covered in the second section. Finally, a taxonomy of hallucination types with their distributions over different setups will be explained. OpenAI GPT-3.5 and GPT-4 models have been used for experiments. All these experiments run by adapting the publicly available RARR repository [55] and unofficial implementation of COVE [56].

5.1 Analysis

In this section, COVE and RARR experiments will be presented with graphs such as barplots, mosaic plots, etc. In addition to these, human evaluation results with 25 participants will also be shared under this section.

5.1.1 Search vs. Context

Modern fact-checkers, despite having complex and logical pipelines, usually struggle to retrieve the relevant context with their self-knowledge, especially on topics that require up-to-date information. Finding the correct context is one of the most critical steps in order to refine a hallucinated claim.

Integrating a search tool is usually the first potential solution to this aforementioned problem. While these search-enhanced fact-checkers aim to fix the issue of retrieval, it is also possible that this information is not available or easy-to-retrieve using search engines.

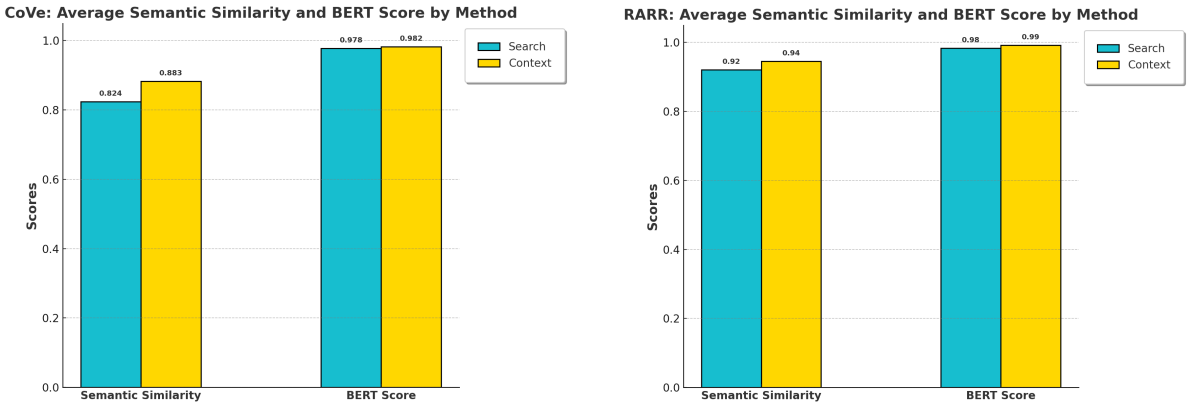
What happens if the retrieval is perfect? This question is one of the main motivations for adding original documents as a knowledge source to existing fact-checking pipelines. SummEdits dataset has been used for the following experiments. In this dataset, there are the following items:

- Gold Summary
- Perturbed (Hallucinated) Summary
- Original Document

where the Original Document is actual source for the Gold Summary. Each experiment runs through the setup that integrates original document as the actual knowledge source. Internal knowledge and search engine tools are disabled. This setup only runs the pipeline with original document, and simulating the scenario in which the retrieval of the correct

knowledge source is flawless. As shown in the following graphs, adding original document as context yields a significant improvement across all metrics.

Impact of Context on Semantic and BERT Scores

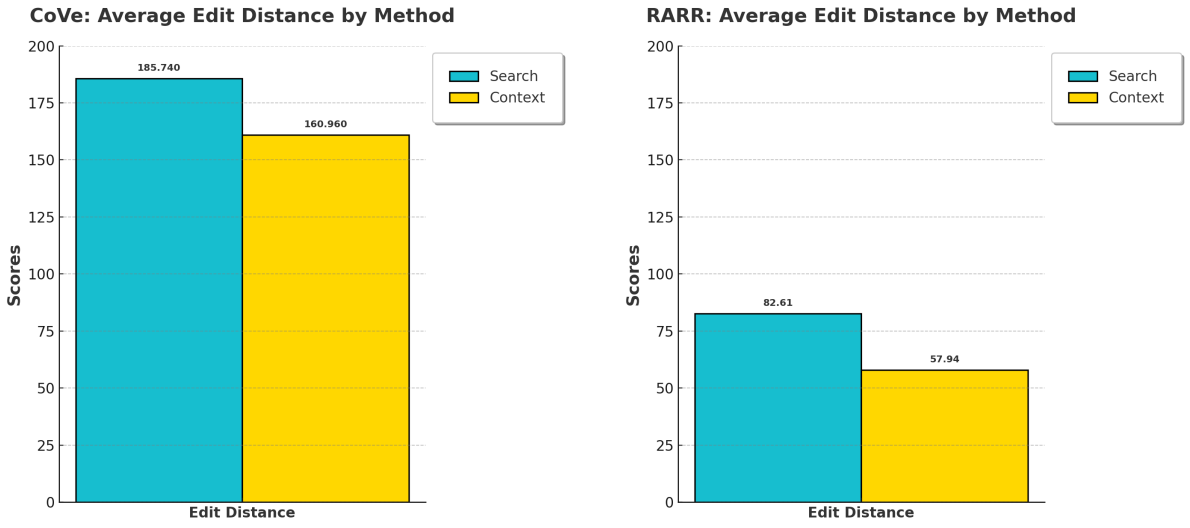


(a) CoVE - Semantic Comparison

(b) RARR - Semantic Comparison

Figure 5.1: Benchmark results on SummEdits dataset. The comparison shows that using context instead of search engines improves both the semantic and BERT scores.

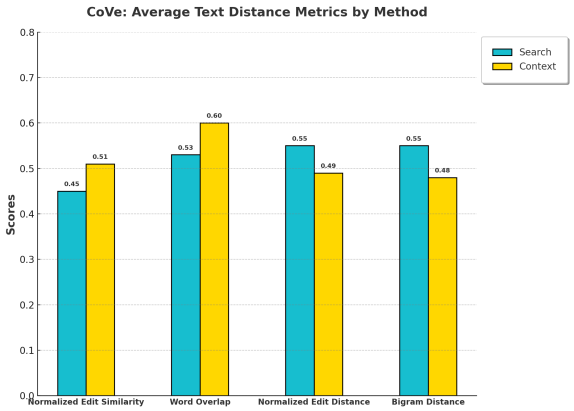
Impact of Context on Text Distance Metrics



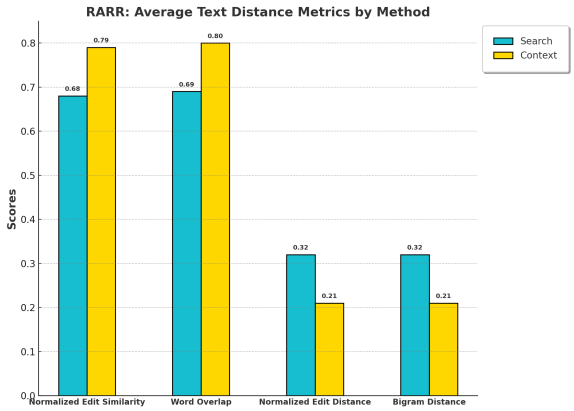
(a) CoVE - Edit Distance

(b) RARR - Edit Distance

Figure 5.2: Edit Distance comparison on SummEdits dataset for CoVE and RARR.



(a) CoVE - Text Distances

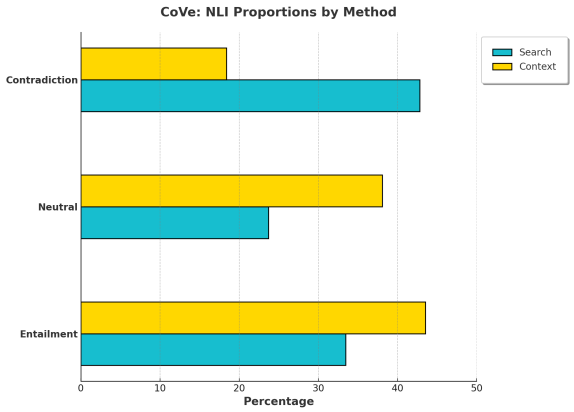


(b) RARR - Text Distances

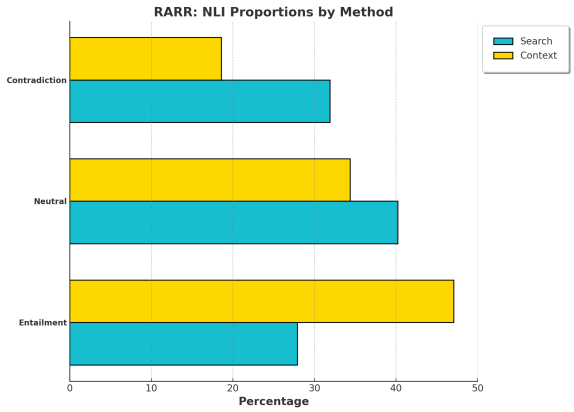
Figure 5.3: Text Distances comparison on SummEdits dataset for CoVE and RARR.

Incorporating context boosts performance, leading to lower edit and bigram distances, along with higher similarity and overlap.

Impact of Context on NLI



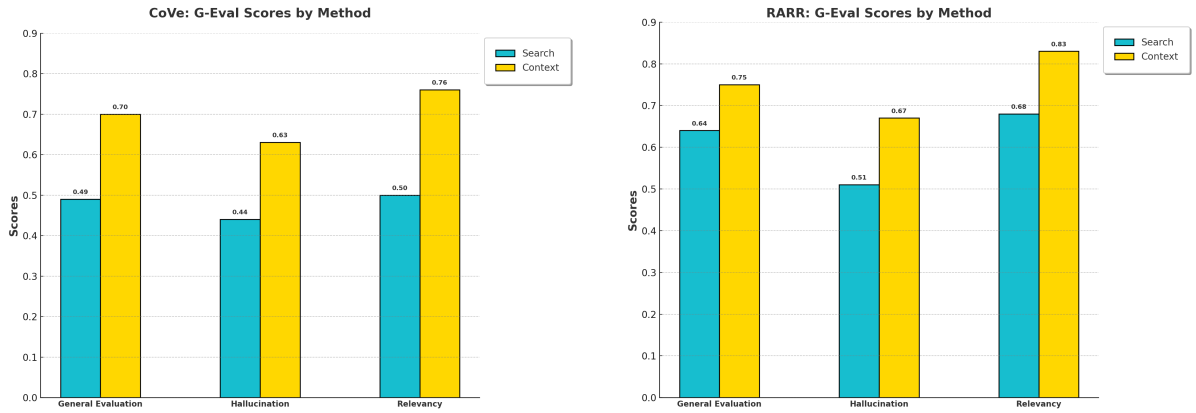
(a) CoVE - NLI Results



(b) RARR - NLI Results

Figure 5.4: Benchmark results on SummEdits dataset for NLI.

Impact of Context on G-Eval



(a) CoVE - G-Eval Comparison

(b) RARR - G-Eval Comparison

Figure 5.5: Benchmark results on SummEdits dataset for G-Eval.

Adding the context information sets the General Evaluation (G-Eval) metric for RARR on SummEdits to 0.75. While this experiment is quite trivial and the results are expected, it is important to highlight the following conclusions:

1. **Essential Role of Source:** Context-enhanced version sets a record across all the experiments in this thesis. Despite numerous attempts and approaches, the original document remains irreplaceable by other techniques, such as external tools.

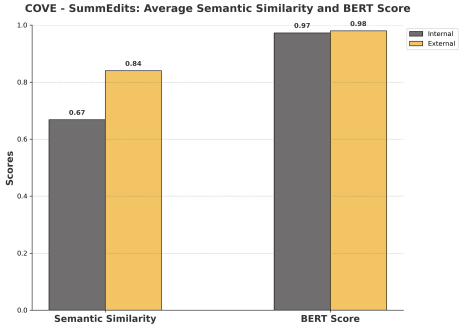
2. **Limitations of Current Pipelines:** Even when the knowledge source is verified and serves as the actual source for a given claim, pipelines still fall short and can not achieve full accurate corrections. This finding encourages future researchers to further explore the pipelines used by fact-checkers and to improve the steps and connections involved.

5.1.2 Self-knowledge vs. Search Engines

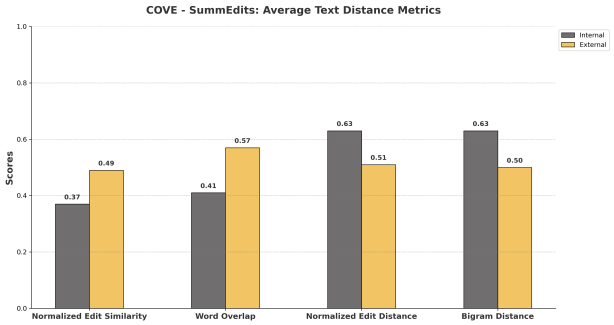
RARR and COVE, two popular fact-checkers in the literature, differ in the way that COVE uses self-knowledge in first released version whereas RARR uses using search engine for information retrieval in order to verify a given user query.

Two datasets used in this thesis, SummEdits and ExpertQA, include samples that may require access to online resources. When asked about information before their knowledge cut-off, LLMs can sometimes retrieve relevant details. Therefore, this section will address the question of whether LLM self-knowledge is sufficient to answer these types of questions. Below charts highlight the differences between search engines (legend is external) and self-knowledge (legend is internal).

SummEdits - COVE

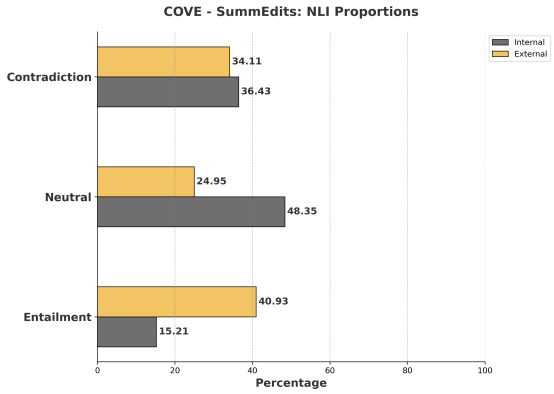


(a) Semantic BERT Scores for SummEdits

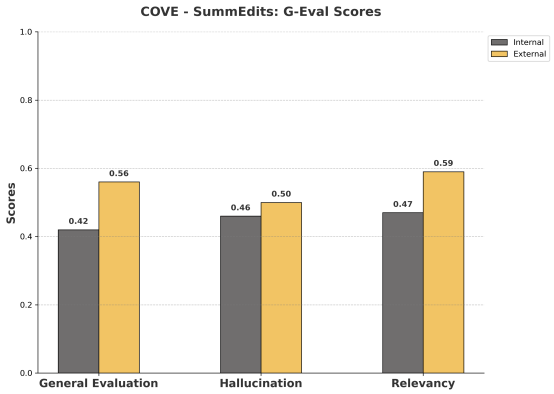


(b) Text Distances for SummEdits

Figure 5.6: Comparison of internal knowledge and search tool for Semantic Similarity and Text Distances.



(a) NLI Proportions for SummEdits

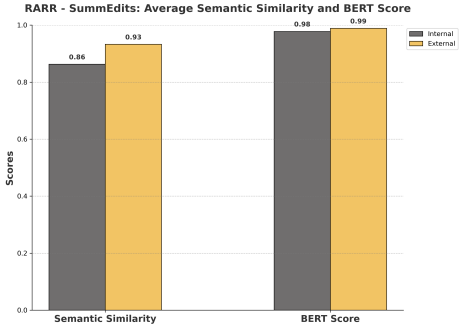


(b) G-Eval Scores for SummEdits

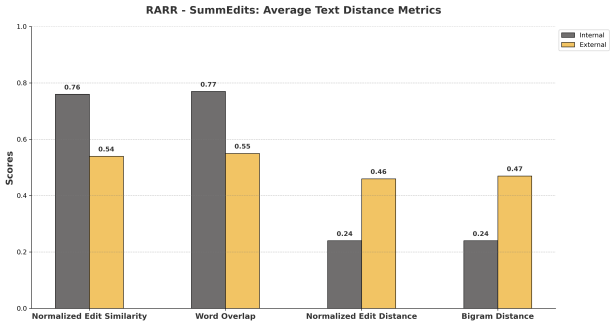
Figure 5.7: Comparison of internal knowledge and search tool for NLI and G-Eval.

COVE metrics show that search engine tool performs far better than self-knowledge for fact-checking on news, with about 14% improvement on General Evaluation and ~%25 increase on NLI-entailment.

SummEdits - RARR

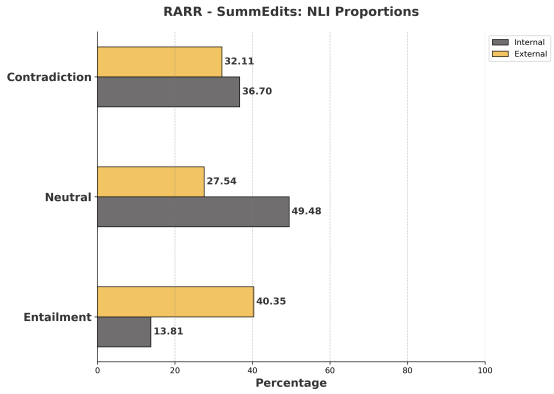


(a) Semantic BERT Scores for SummEdits

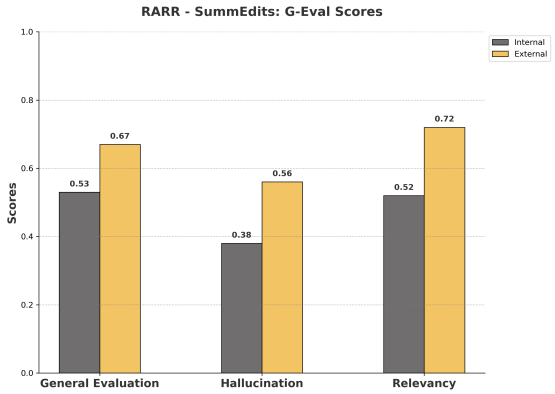


(b) Text Distances for SummEdits

Figure 5.8: Comparison of internal knowledge and search tool for Semantic Similarity and Text Distances.



(a) NLI Proportions for SummEdits

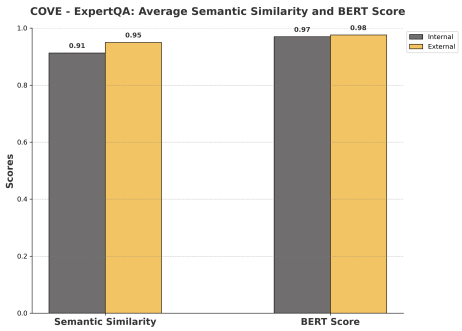


(b) G-Eval Scores for SummEdits

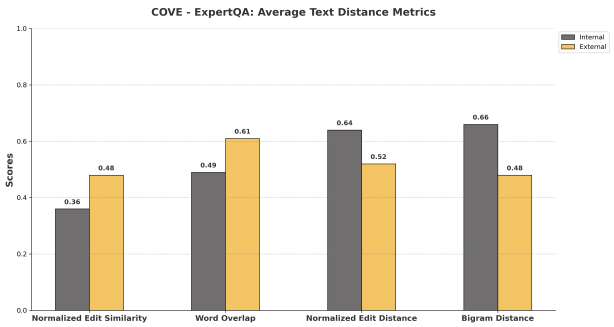
Figure 5.9: Comparison of internal knowledge and search tool for NLI and G-Eval.

RARR's results show that using search engines works better for fact-checking news than self-knowledge. It has a 14% boost in General Evaluation and about a 27% increase in NLI entailment.

ExpertQA - COVE

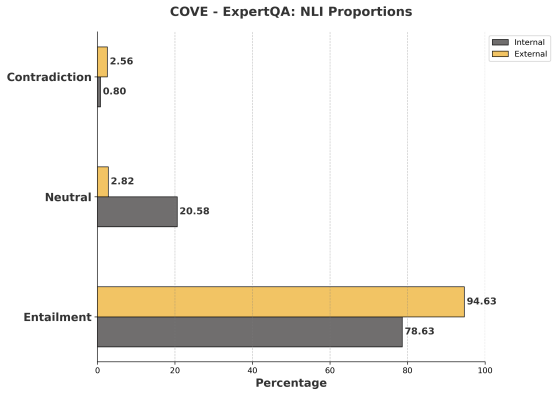


(a) Semantic BERT Scores for ExpertQA

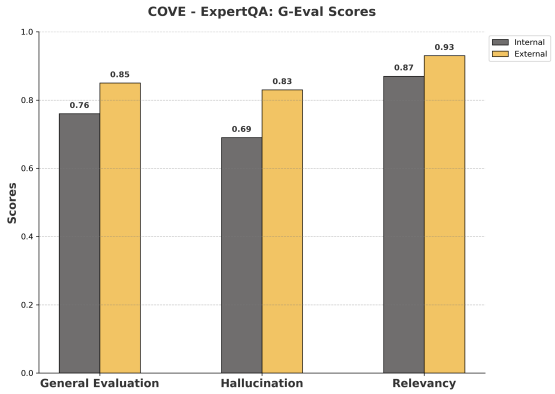


(b) Text Distances for ExpertQA

Figure 5.10: Comparison of internal knowledge and search tool for Semantic Similarity and Text Distances.



(a) NLI Proportions for ExpertQA

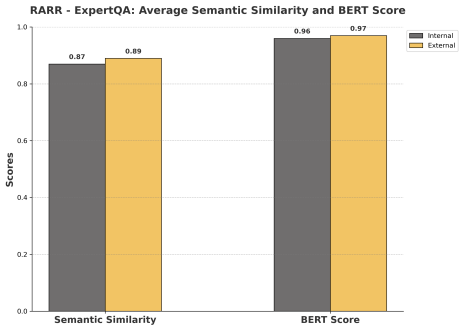


(b) G-Eval Scores for ExpertQA

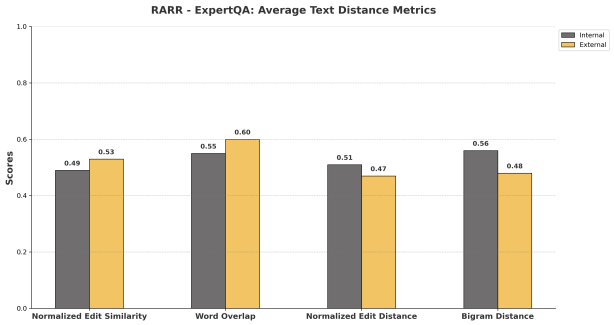
Figure 5.11: Comparison of internal knowledge and search tool for NLI and G-Eval.

Similar to the findings and conclusions drawn from SummEdits, COVE demonstrates a significant improvement of ~11% in General Evaluation by utilizing search engines instead of relying only on internal knowledge in ExpertQA dataset as well.

ExpertQA - RARR

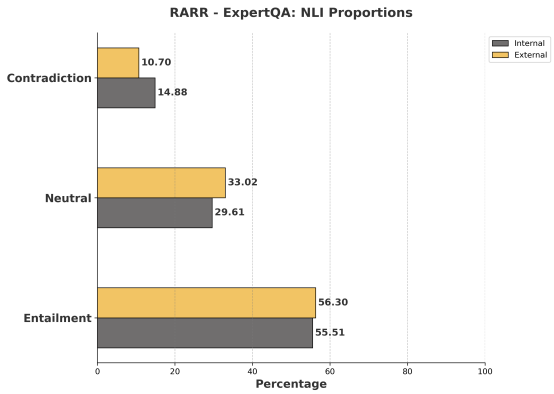


(a) Semantic BERT Scores for ExpertQA

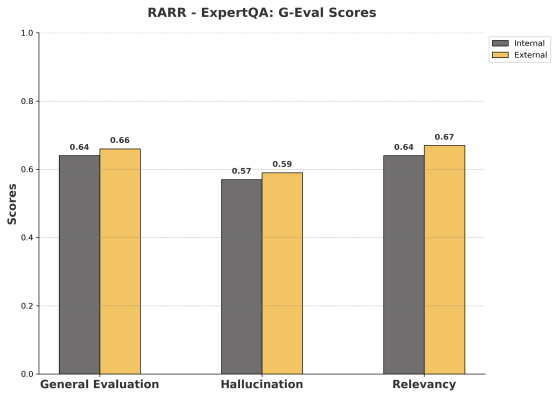


(b) Text Distances for ExpertQA

Figure 5.12: Comparison of internal knowledge and search tool for Semantic Similarity and Text Distances.



(a) NLI Proportions for ExpertQA



(b) G-Eval Scores for ExpertQA

Figure 5.13: Comparison of internal knowledge and search tool for NLI and G-Eval.

While RARR's improvement is not as large as expected, it still demonstrates that external methods outperform internal ones.

Therefore, it is generally advisable to use search engines for fact-checking information that requires real-time access or any type of up-to-date data. This approach can boost factuality and relevance of the information verified.

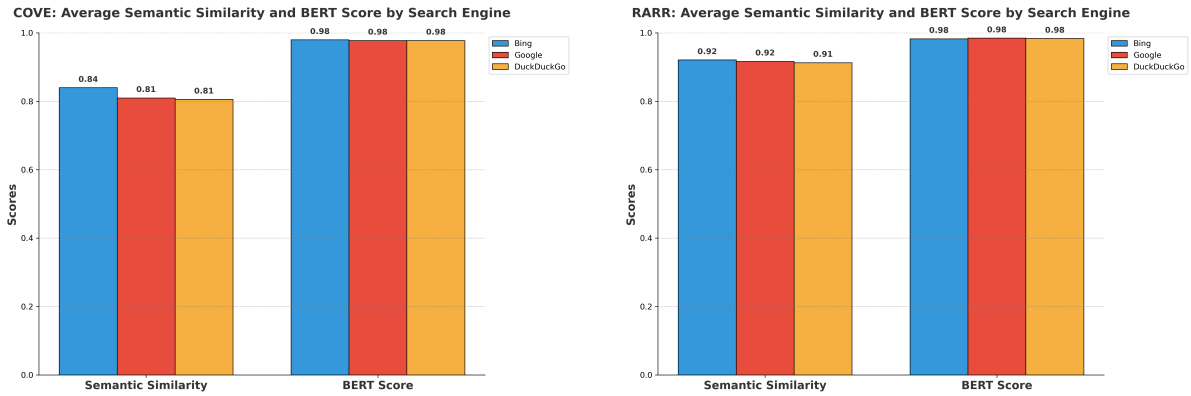
5.1.3 Different Search Engines (Bing, Google, DuckDuckGo)

Web search engines are currently opening-doors to the most up-to-date knowledge available on the internet. In some cases where knowledge source is fixed, such as official reports and internal or confidential documentations, search engines might not be an appropriate solution

for fact-checkers. However, in most situations, fact-checking pipelines rely on using external tools, particularly search engines. Using search engines improves factuality and relevance, as shown in previous subsection.

Selecting the right search engine is also quite important because it might affect how accurate, reliable the information gathered is. Different search engines focus on different sources and use their own underlying implementations, algorithms, or tricks. This can definitely change the results. To explore which search engine works best for fact-checking, following sub-subsections present bar plots comparing Bing, Google, and DuckDuckGo.

Impact of Different Search Engines on Semantic and BERT Scores



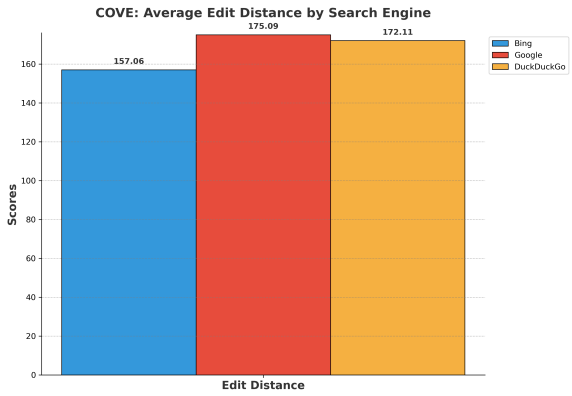
(a) CoVE - BERT Semantic Comparison

(b) RARR - BERT Semantic Comparison

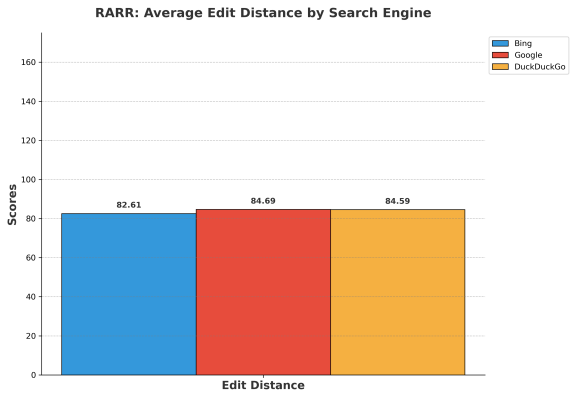
Figure 5.14: Benchmark results on SummEdits dataset for BERT Semantic Comparison.

Using the default setup with both fact-checking methods, different behaviors have been observed across COVE and RARR.

Impact of Different Search Engines on Text Distance Metrics

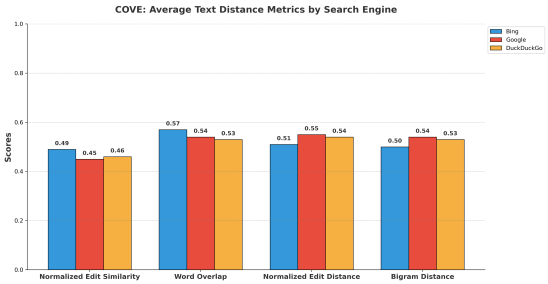


(a) CoVE - Edit Distance

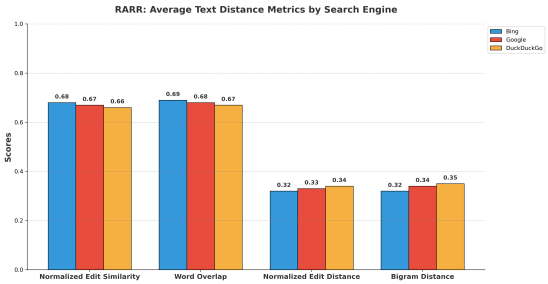


(b) RARR - Edit Distance

Figure 5.15: Edit Distance comparison on SummEdits dataset using different search engines for CoVE and RARR.



(a) CoVE - Text Distances

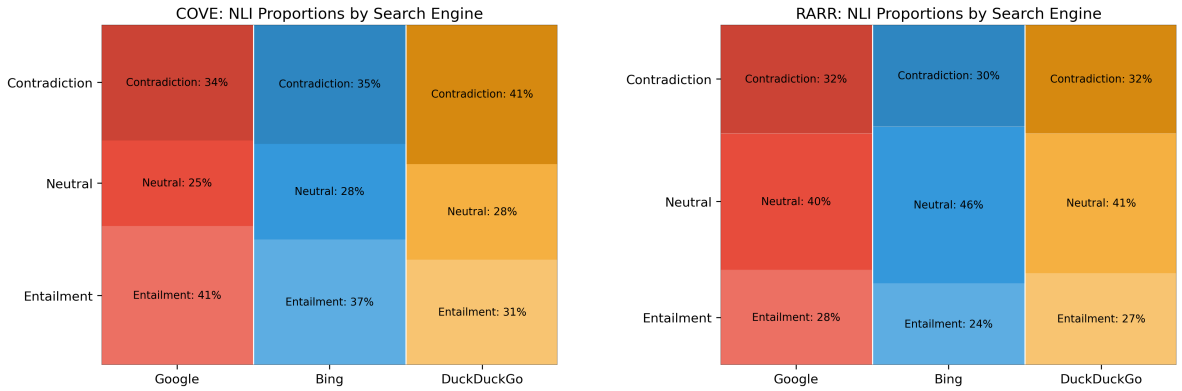


(b) RARR - Text Distances

Figure 5.16: Text Distances comparison on SummEdits dataset using different search engines for CoVE and RARR.

When checking edit distance and text distance metrics (ROUGE-1, ROUGE-2, and normalized edit distance), COVE performs better with Bing, while RARR results remain largely consistent.

Impact of Different Search Engines on NLI

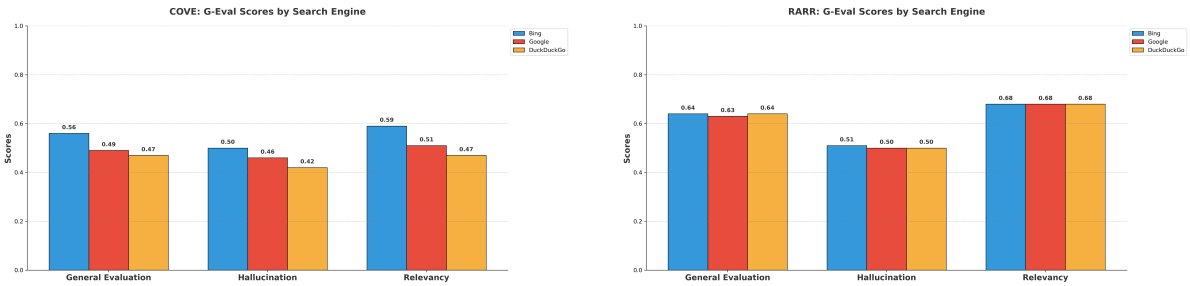


(a) Search Engines - NLI Comparison

(b) Search Engines - NLI Comparison

Figure 5.17: Benchmark results on SummEdits dataset for NLI using different search engines. Contextual information in NLI tasks shows varying performance across search engines.

Impact of Different Search Engines on G-Eval



(a) COVE Search Engines - G-Eval Comparison

(b) COVE Search Engines - G-Eval Comparison

Figure 5.18: Benchmark results on SummEdits dataset for G-Eval using different search engines.

RARR results are almost identical and fall within <2% (also reported in original paper [36] as <1%). In contrast, COVE shows quite a bit of fluctuation, yielding different results for each search engine. Here, enhanced version of COVE (with Search Engine support) uses search snippets, while RARR uses passages that involve ranking and retrieving chunks from respective websites.

5.1.4 Human Evaluation

The main goal of this human evaluation is to judge LLM-as-a-judge (G-Eval) and NLI, exploring the correlation between human and machine evaluation methods.

Study Format and Instructions

User study was conducted with 25 participants. Each user is provided with 10 correct summaries and associated 4 questions. In total, each user replied to 40 questions, and in total 1000 questions were answered.

Following instructions are provided at the beginning of the form:

> Instructions
> Read the correct summary first.
> Compare the correct summary with Summary A and Summary B.
> There are no right or wrong answers.
> Both summaries can be good or bad.
> For each summary (A and B), there are two types of questions:
>
> 1. Choose the option that best fits the blank:
>   - **Contradicts**: Disagrees with the correct summary
>   - **Supports**: Agrees with the correct summary
>   - **Partially aligns with**: Only somewhat related or unrelated
> 2. Rate Quality (Factual accuracy + Relevance):
>   - **Factual accuracy**: Is it based on facts? Avoids misinformation?
>   - **Relevance**: Does the summary cover the main points? Not off-topic?

Then, users were asked to evaluate each question. The first question represents NLI results, whereas the second question corresponds to G-Eval General Evaluation metric.

In each question, we include samples from RARR or COVE as either summary A or B. Correct summary represents the ground truth from SummEdits dataset. Summary A or Bs from fact-checkers are generated using Bing search engine snippets. Both fact-checkers were provided with the same hallucinated version of the correct summary and asked to mitigate hallucinations. In 5.19, a sample screenshot from the evaluation form is provided.

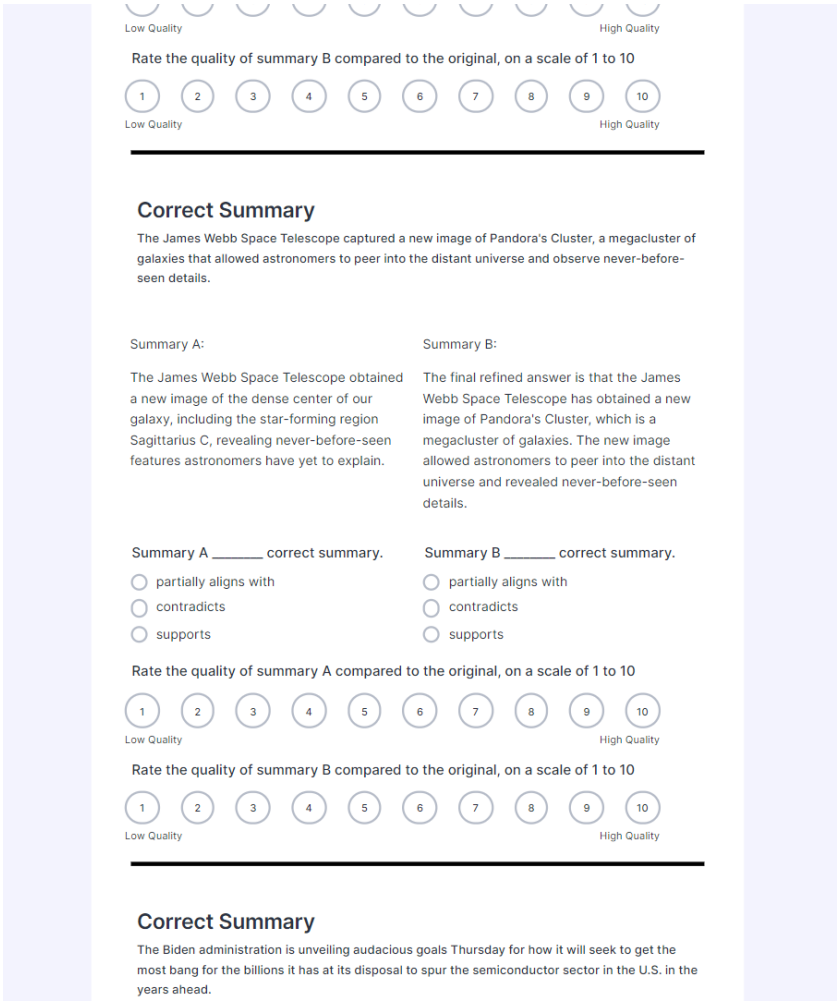


Figure 5.19: A screenshot from Human Evaluation Form.

Users have been informed that this thesis is studying how well AI-generated summaries match correct information and results will help improve evaluation of AI systems. All responses collected are anonymous and used only for this thesis.

Quality (Factual accuracy + Relevance) Results

To evaluate the alignment between the G-Eval scores and human evaluations for the RARR and COVE methods, we analyzed the mean scores and their differences. Results are summarized in Table 5.1.

Method	Human Mean Score	G-Eval Score	Diff
RARR	0.68	0.65	0.03
COVE	0.54	0.52	0.02

Table 5.1: Alignment between G-Eval scores and human evaluations.

For RARR, average human score is 0.68, and average G-Eval score is 0.65. For the COVE method, average human score is 0.54, and average G-Eval score is 0.52. G-Eval scores are slightly lower than human evaluations.

These minor differences for both RARR and COVE suggest that G-Eval scores closely reflect human evaluations for both methods, with a deviation of ±3%.

Natural Language Inference Results

We also compared human evaluation and ground truth (GT) values for Natural Language Inference (NLI) across three categories: Entailment, Neutral, and Contradiction. As discussed before, DeBERTaV3 model [48] is used for NLI evaluation. The results are presented in Table 5.2.

Method	Human			NLI Model		
	Entailment	Neutral	Contradiction	Entailment	Neutral	Contradiction
RARR	45	40	15	30	49	21
COVE	31	37	32	28	47	25

Table 5.2: Comparison of Human Evaluation and NLI predictions

In both methods, there is a higher percentage of Entailment in human evaluations compared to the NLI model, particularly in RARR. Also, percentage of Neutral instances is lower in human evaluations. NLI model (DeBERTaV3), is more likely to classify instances as Neutral. Contradiction shows higher percentages in human evaluations for COVE compared to the NLI model. Overall, as demonstrated by evaluation of experiments and human evaluation, RARR performs better than COVE in SummEdits dataset.

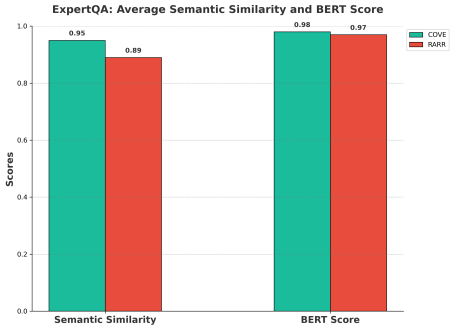
Alignment

Analyses indicate a strong alignment between G-Eval scores and human evaluations for both RARR and COVE methods in rating quality. NLI results show small differences between human evaluations and actual answers, with dominant classes mostly the same. This consistency means that G-Eval is a reliable tool for approximating human assessments. It can be used in scenarios where human evaluations are impractical when there are time or resource constraints.

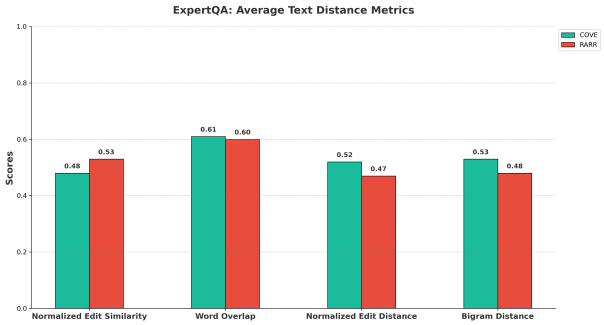
5.1.5 Performance on ExpertQA Dataset

Following SummEdits, ExpertQA results are also explored, it is longer than SummEdits shorter then longfact but contains a corrected text by human experts allowing better evaluation, using the same pipeline for SummEdits.

Following plots provide an overview of results:

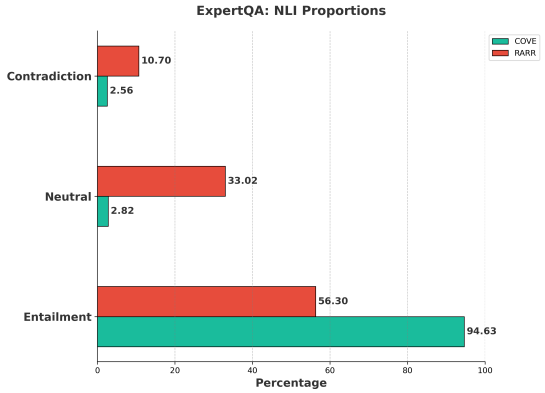


(a) Semantic BERT Scores for ExpertQA

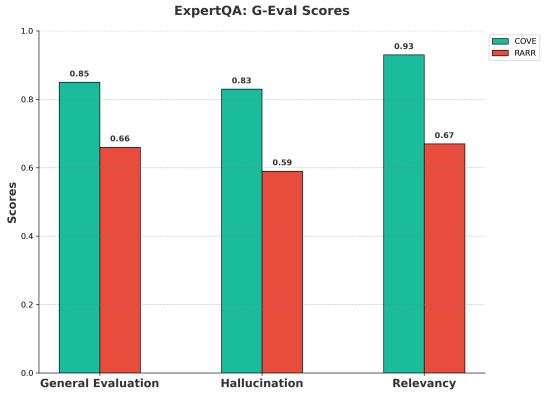


(b) Text Distances for ExpertQA

Figure 5.20: Comparison of internal knowledge and search tool for Semantic Similarity and Text Distances.



(a) NLI Proportions for ExpertQA



(b) G-Eval Scores for ExpertQA

Figure 5.21: Comparison of internal knowledge and search tool for NLI and G-Eval.

Method	Correction	Original Text
COVE	The Delivery Performance Rate (DPR) in transportation management is calculated by dividing the number of successful deliveries by the total number of deliveries attempted, and then multiplying by 100 to get a percentage. The total number of deliveries includes all deliveries made within a specified period, whether they were on time or late. The resulting DPR value indicates the effectiveness and efficiency of transportation management in fulfilling its delivery commitments.	The delivery quota in transportation management can be calculated using the Delivery Performance Rate (DPR) formula, which measures the percentage of on-time deliveries against the total number of deliveries made within a specific period. The formula for calculating DPR is: DPR = (On-time deliveries / Total number of deliveries) x 100. In this formula, "on-time deliveries" refers to the number of deliveries that meet the established delivery window, while the "total number of deliveries" includes all deliveries made within the specified period, whether they were on time or late. The resulting DPR value indicates the effectiveness and efficiency of the transportation management in fulfilling its delivery commitments.
RARR	The delivery quota in transportation management can be calculated by adding up the total time for all deliveries and dividing by the number of deliveries. This metric can be used as a tool for negotiation where better pricing could be offered for a more realistic time of delivery.	
COVE	Apple and Jimmy Iovine's efforts to curb the onset of digitally pirated music can be considered successful as [...] a decline in music piracy rates.	Apple and Jimmy Iovine's efforts to curb the onset of digitally pirated music can be considered successful, as [...] a decline in music piracy rates.
RARR	The information provided in your statement about Apple and Jimmy Iovine's efforts to reduce music piracy is incorrect.	

Table 5.3: ExpertQA: Comparison of COVE and RARR Corrections to the Original Texts

In first example from Table 5.3, RARR describes the delivery quota calculation as "adding up the total time for all deliveries and dividing by the number of deliveries", but it should be on-time deliveries. This explanation fundamentally changes the concept from the original. COVE's correction, despite having minor issues, still retains the fundamental concept of calculating a percentage based on successful versus total deliveries.

In second example, despite having a long-text answer here in Table 5.3, RARR correction refuses to elaborate in many cases. It simply states that the information is incorrect without providing specific details as seen with Apple example.

To sum up, for ExpertQA long-form dataset, zero-shot setup performs better than few-shot setup.

5.1.6 Performance on LongFact Dataset

3rd dataset used within this thesis, called LongFact, does not contain gold answers. For that reason, a different evaluation framework, FActScore [29], has been used for evaluations on LongFact dataset.

In publicly available GitHub repository, LongFact already provides concept and object questions. For each subject (e.g. architecture, chemistry, etc), there are 30 questions provided. Each of these questions is intended to produce long answers when asked to LLMs. To give a clearer idea of length of typical long-form factual answers generated in this work, please refer to Table 5.4.

Metric	Character Count	Word Count
Mean	1758	244
Standard Deviation	517	70
Median	1838	260

Table 5.4: LLM Response Lengths for LongFact questions (`max_tokens = 500`)

On average, each response contains 244 words, which is ~3-5 paragraphs. These long-form text responses generated by LLMs, are then passed to fact-checking pipelines, to COVE and RARR. These pipelines are then expected to correct any hallucinations that exist in the generated long-form answer.

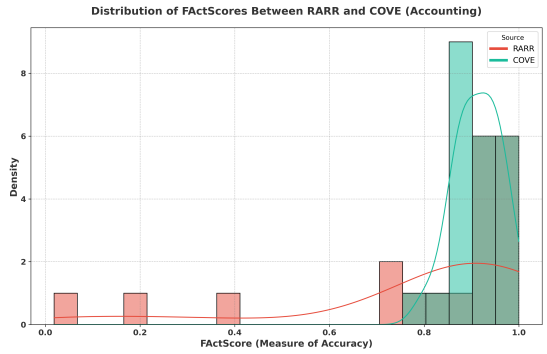
After corrections, the final refined answer is obtained. Evaluation factuality and relevancy of these long answers is quite challenging due to the fact there are no gold answers in this dataset.

To evaluate these results, FActScore [54] has been used. It gives a final score that is a percentage of factual atomic facts. Factuality is determined by the knowledge source. FActScore is an extensible evaluation package, that allows registering custom knowledge sources. More detail on FActScore setup can be found under 4-Methodology chapter. Out of 39 topics proposed in LongFact dataset, following 4 topics have been selected: Accounting, Astronomy, Gaming, and Physics.

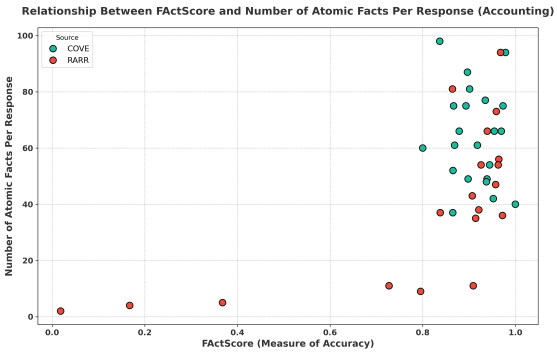
Results are shown in four different plots: (1) a histogram with KDE that shows how FactScores are spread out, (2) a scatter plot that shows link between FactScore and number of atomic facts, (3) a boxplot that highlights how many atomic facts are in each response, and (4) a barplot that looks at supported versus unsupported atomic facts. These plots for all subjects can be found in the upcoming subsections.

LongFact - Accounting

This dataset covers key accounting topics like fiscal years, revenue recognition, depreciation, and cost accounting.

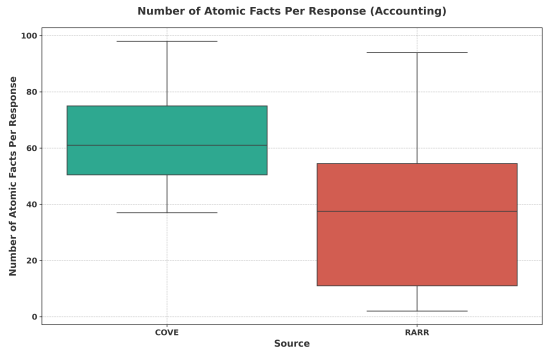


(a) FactScore Distribution

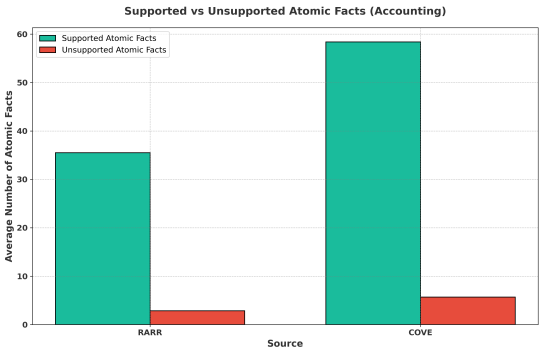


(b) FactScore vs. Atomic Facts per Response

Figure 5.22: (a) Distribution of FactScores, showing the spread of scores across the dataset. (b) Relationship between FactScore and the number of atomic facts per response.



(a) Number of Atomic Facts per Response

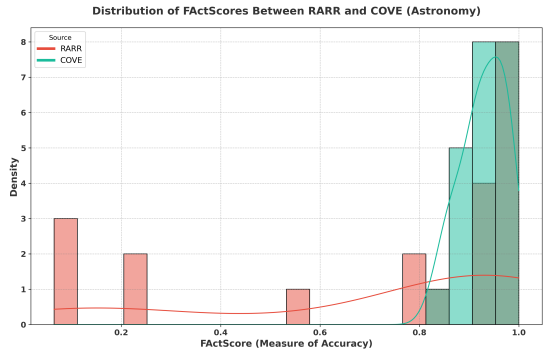


(b) Supported vs. Unsupported Atomic Facts

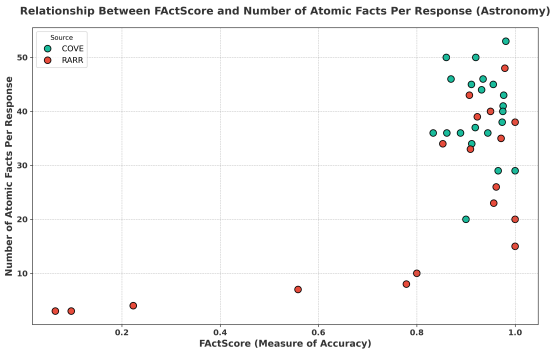
Figure 5.23: (a) Number of atomic facts present in each response, showing that COVE leads (b) Comparison between supported and unsupported atomic facts

LongFact - Astronomy

This dataset includes questions about different astronomy topics, like gravitational lensing, redshift, cosmic inflation, neutron stars, and exoplanets.

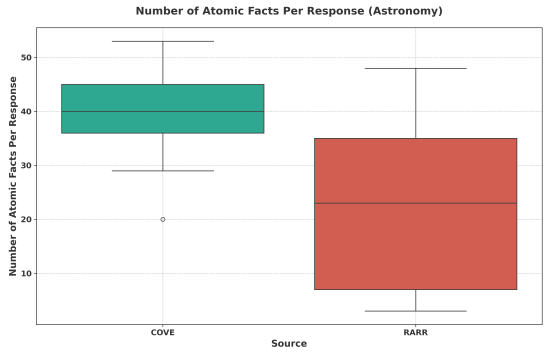


(a) FactScore Distribution

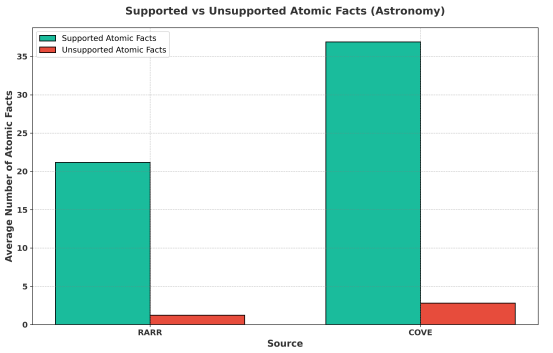


(b) FactScore vs. Atomic Facts per Response

Figure 5.24: (a) Distribution of FactScores, showing the spread of scores across the dataset. (b) Relationship between FactScore and the number of atomic facts per response.



(a) Number of Atomic Facts per Response

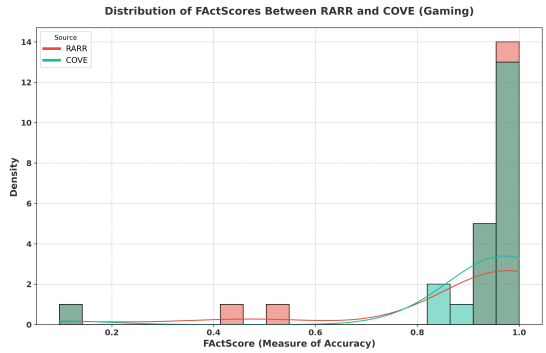


(b) Supported vs. Unsupported Atomic Facts

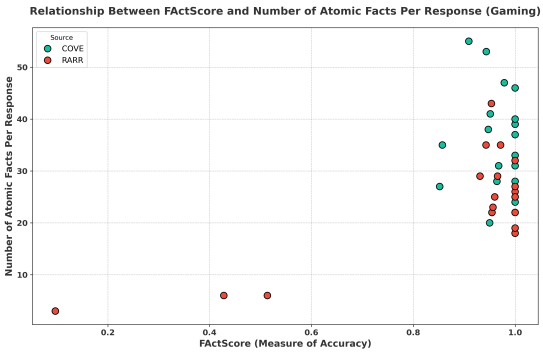
Figure 5.25: (a) Number of atomic facts present in each response, showing that COVE leads (b) Comparison between supported and unsupported atomic facts

LongFact - Gaming

This dataset contains questions about the gaming industry, covering topics like AI in game design, player agency, cross-platform gaming, VR/AR, and monetization practices.

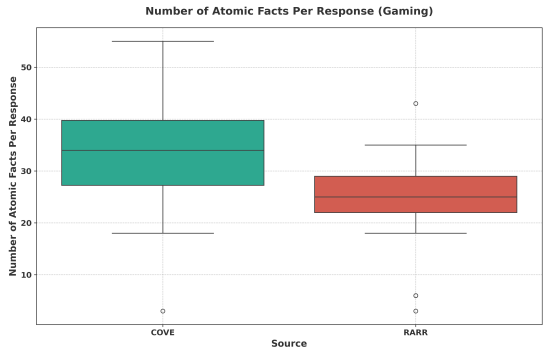


(a) FactScore Distribution

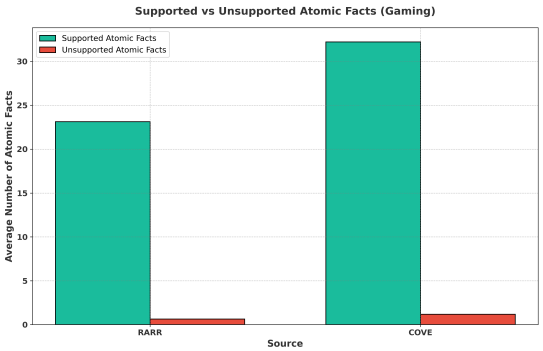


(b) FactScore vs. Atomic Facts per Response

Figure 5.26: (a) Distribution of FactScores, showing the spread of scores across the dataset. (b) Relationship between FactScore and the number of atomic facts per response.



(a) Number of Atomic Facts per Response

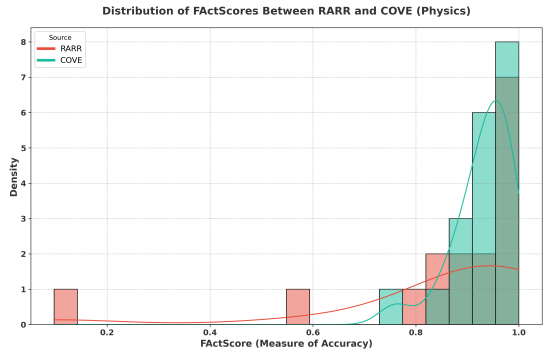


(b) Supported vs. Unsupported Atomic Facts

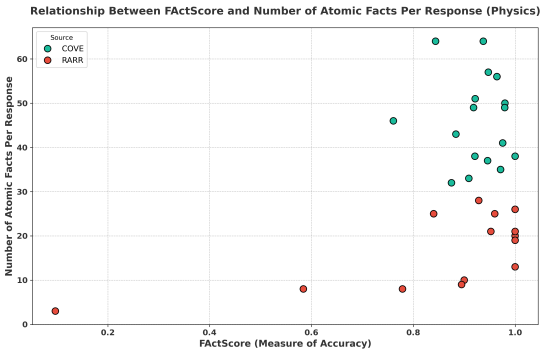
Figure 5.27: (a) Number of atomic facts present in each response, showing that COVE leads (b) Comparison between supported and unsupported atomic facts

LongFact - Physics

This dataset includes questions on various advanced physics topics, such as quantum mechanics, special relativity, chaos theory, quantum field theory, and more.

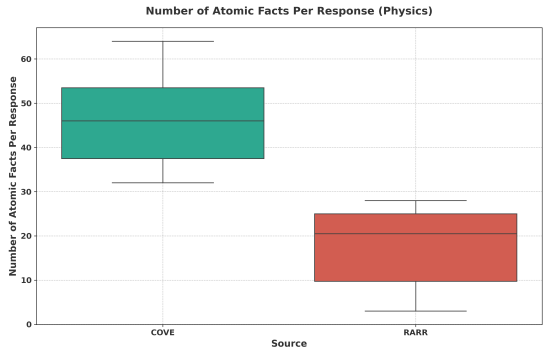


(a) FactScore Distribution

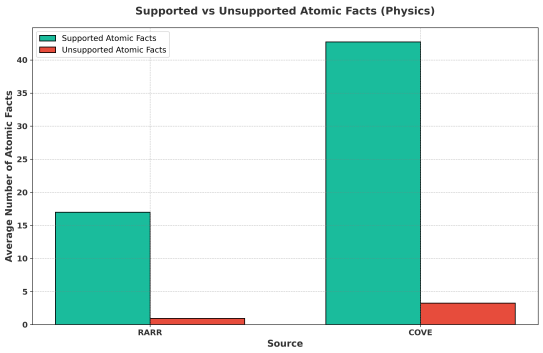


(b) FactScore vs. Atomic Facts per Response

Figure 5.28: (a) Distribution of FactScores, showing the spread of scores across the dataset. (b) Relationship between FactScore and the number of atomic facts per response.



(a) Number of Atomic Facts per Response



(b) Supported vs. Unsupported Atomic Facts

Figure 5.29: (a) Number of atomic facts present in each response, showing that COVE leads (b) Comparison between supported and unsupported atomic facts

In each category, including astronomy, gaming, and physics, COVE is more reliable and is able to provide longer answers. Having longer answers allows FactScore to generate more atomic facts from LLM-generated responses. With longer text and more atomic facts, there is more room for error but COVE was able to provide factual information according to the knowledge source.

Based on the experiments, the main conclusion is that COVE outperforms RARR on the Long-Fact dataset. Interestingly, in previous tests, RARR had performed better than COVE. One of the reason is that RARR uses short few-shot examples by default, while COVE, in its default setup, uses zero-shot. This gives COVE more flexibility to adapt to the longer text format. For these type of 2-3 paragraph datasets, if few-shot used, then examples need to be adjusted to reflect dataset format.

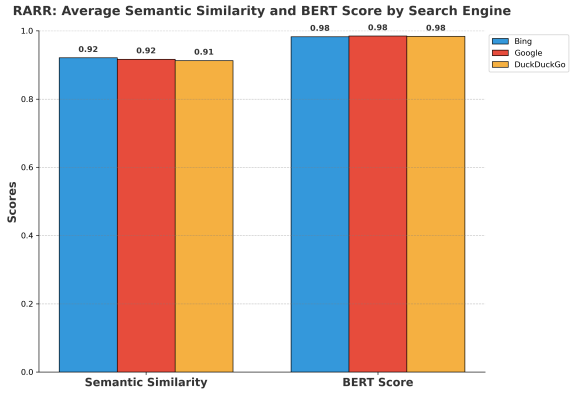
5.2 Improvements

As an additional contribution, this thesis aims to address some of the shortcomings of existing methods by introducing specific improvements into the pipelines. These enhancements include the use of multiple search engines for retrieval, various prompts and prompting techniques (e.g. chain-of-thought and more few-shot examples), and the use of focused-context and specificity prompts.

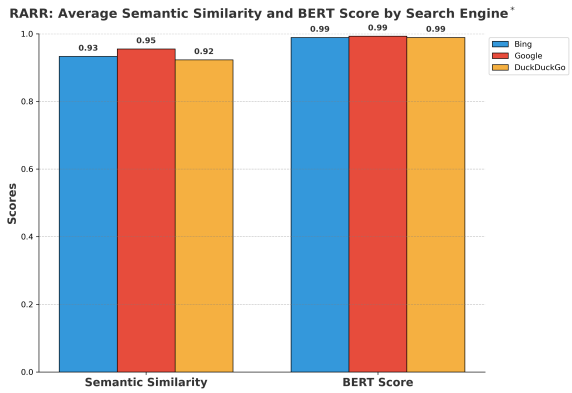
5.2.1 Search Snippets over Passages

In the following experiments, Bing search full-text retrieval from URLs and finding relevant passages are referred to as passages, whereas Bing search snippets are referred to as snippets. Main goal is to compare how two different types of search engine usage affect fact-checking accuracy, detailed analysis with plots is provided in the following subsection.

Impact of Passage Retrieval vs. Snippets on Semantic and BERT Scores



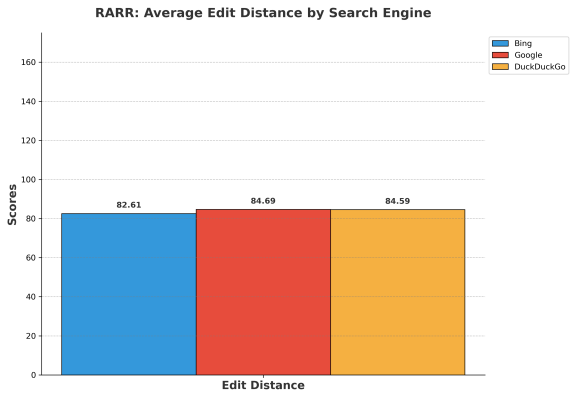
(a) Passage Retrieval - BERT Semantic Comparison



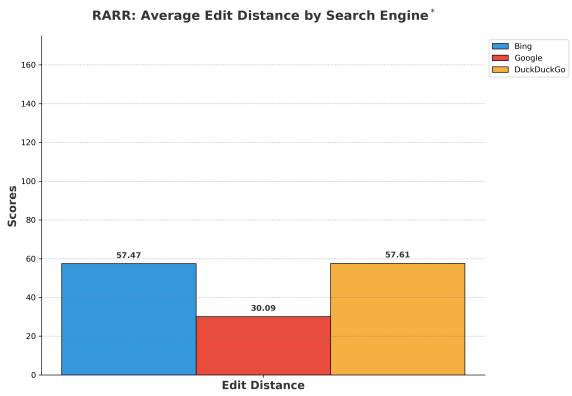
(b) Snippets - BERT Semantic Comparison

Figure 5.30: Benchmark results on SummEdits dataset for BERT Semantic Comparison between passage retrieval and snippets.

Impact of Passage Retrieval vs. Snippets on Text Distance Metrics

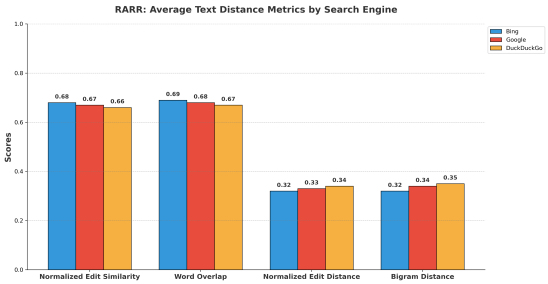


(a) Passage Retrieval - Edit Distance

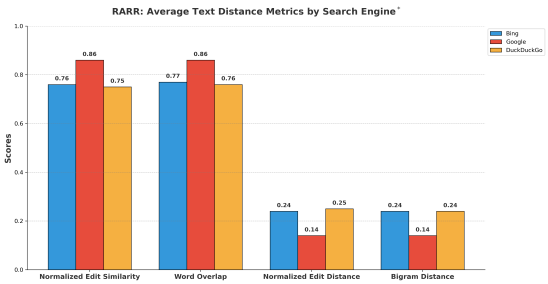


(b) Snippets - Edit Distance

Figure 5.31: Edit Distance comparison on SummEdits dataset using passage retrieval vs. snippets.



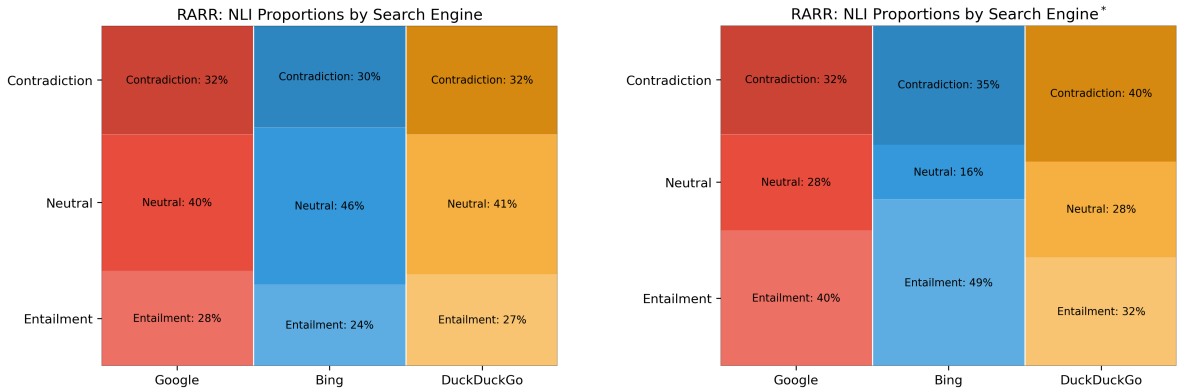
(a) Passage Retrieval - Text Distances



(b) Snippets - Text Distances

Figure 5.32: Text Distances comparison on SummEdits dataset using passage retrieval vs. snippets.

Passage Retrieval vs. Snippets on NLI

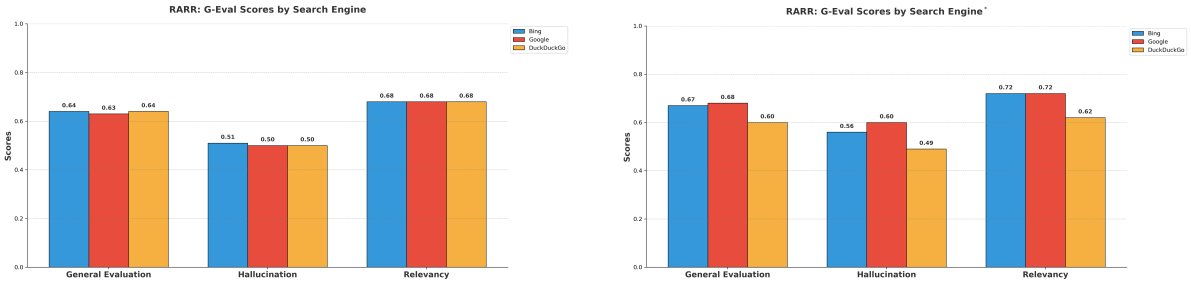


(a) Passage Retrieval - NLI Comparison

(b) Snippets - NLI Comparison

Figure 5.33: Benchmark results on SummEdits dataset for NLI using passage retrieval vs. snippets.

Passage Retrieval vs. Snippets on G-Eval



(a) Passage Retrieval - G-Eval Comparison

(b) Snippets - G-Eval Comparison

Figure 5.34: Benchmark results on SummEdits dataset for G-Eval using passage retrieval vs. snippets.

Snippets are around 8% more effective than passages for fact-checking. While passages provide more context, snippets—despite their brevity—are often more efficient at quickly delivering relevant information. This simplification and shorter context also help LLMs to detect and correct hallucinations better.

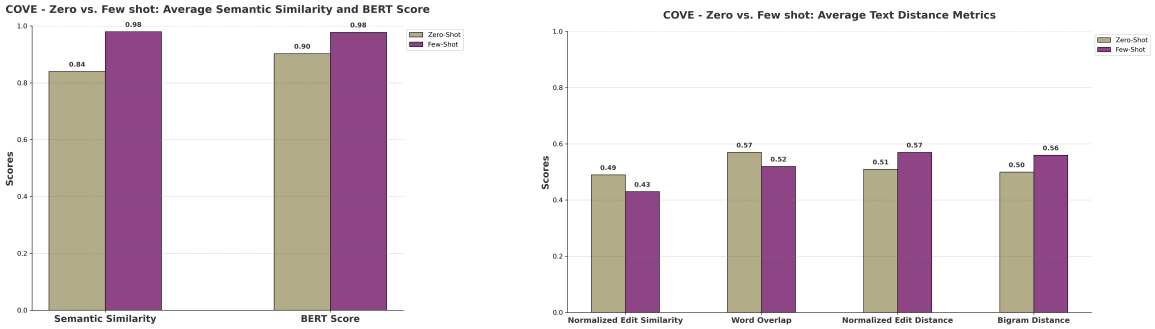
A noticeable takeaway is that DuckDuckGo snippets are not as reliable as those from Google and Bing. DuckDuckGo search snippets tend to repeat instead of adding more context. However, DuckDuckGo does manage to retrieve relevant websites (returning same URLs). When URLs are retrieved and passages are extracted independently, the results are almost

identical. However, if out-of-the-box search engine snippets are used, DuckDuckGo falls short. Its major advantage is that there are no costs associated with screen scraping DuckDuckGo.

5.2.2 Few-shot over Zero-Shot

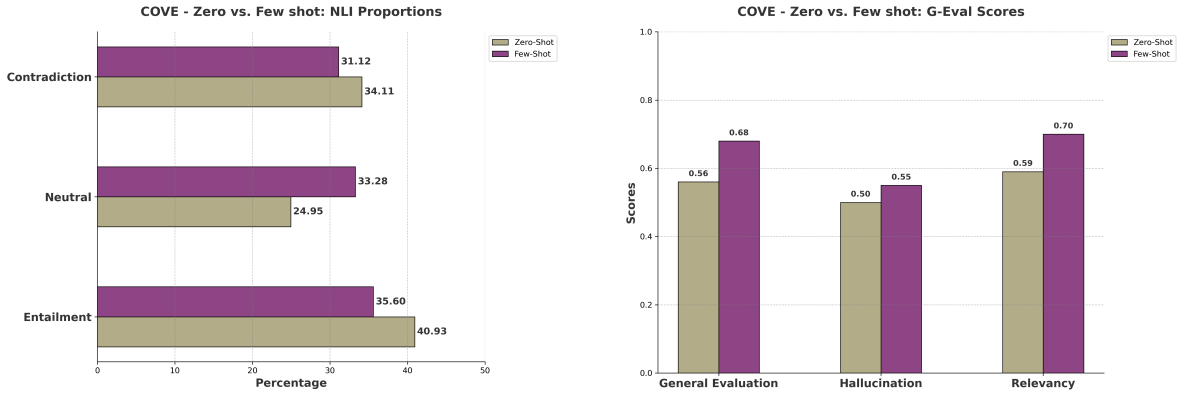
Summedits

In the following experiments comparing few-shot and zero-shot setups, Bing search and snippets are fixed, and dataset used is SummEdits.



(a) Semantic BERT Scores Comparison (b) Text Distances Comparison

Figure 5.35: Comparison of Zero-shot and Few-shot COVE for Semantic Similarity and Text Distances.



(a) NLI Proportions Comparison (b) G-Eval Scores Comparison

Figure 5.36: Comparison of Zero-shot and Few-shot COVE for NLI and G-Eval.

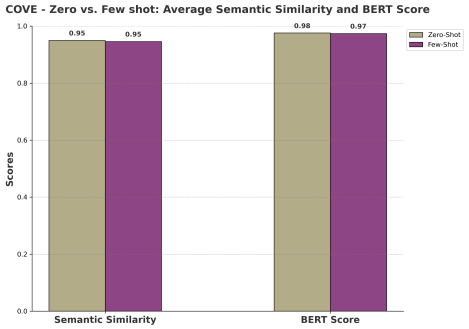
For COVE, adding few-shot examples boosts factuality and shows that the pipeline itself is not quite different than RARR, achieving a similar G-Eval General Evaluation score. It also fixes the error with the pipeline that is starting the answers with an indicator: "The final refined answer is", as shown in Table 5.5 below:

Original	Correction (Zero-Shot)	Improved Correction (Few-Shot)
A Chinese mine collapse killed at least two people and left over 50 others missing. Chinese officials say numerous vehicles were also buried in the collapse.	**The final refined answer is:** A Chinese mine tragedy killed at least two people and left over 50 others missing. Chinese officials say numerous vehicles were also buried in the collapse. **The number of people reported missing after the incident is not provided in the given context.**	The Chinese mine tragedy led to the deaths of at least two individuals, with over 50 people reported missing. Chinese officials have also confirmed that multiple vehicles were buried in the collapse.

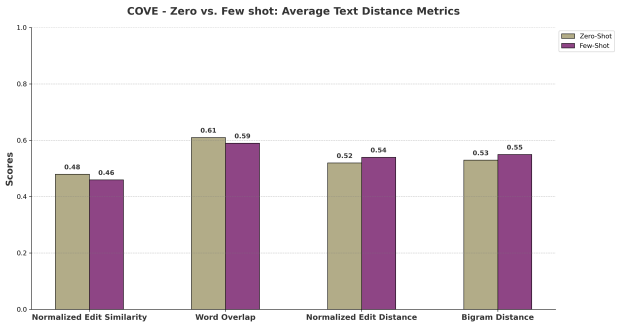
Table 5.5: Comparison of Original and Corrected Texts. Bold Highlighted Part is Redundant.

ExpertQA

In this part, the setup is the same as the previous setting, except the dataset switch from SummEdits to ExpertQA.

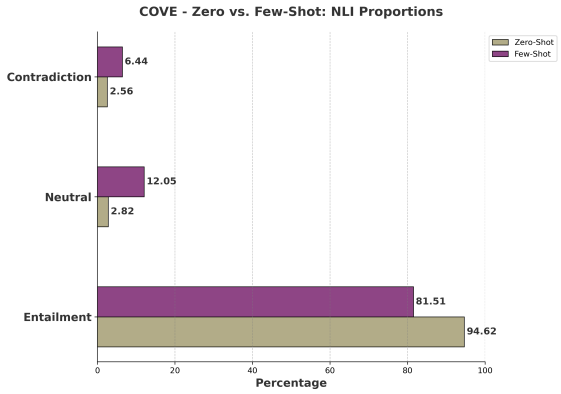


(a) Semantic BERT Scores for ExpertQA

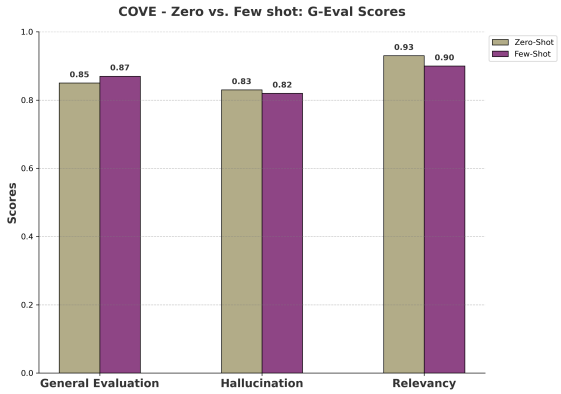


(b) Text Distances for ExpertQA

Figure 5.37: Comparison of Zero-shot and Few-shot ExpertQA for Semantic Similarity and Text Distances.



(a) NLI Proportions for ExpertQA



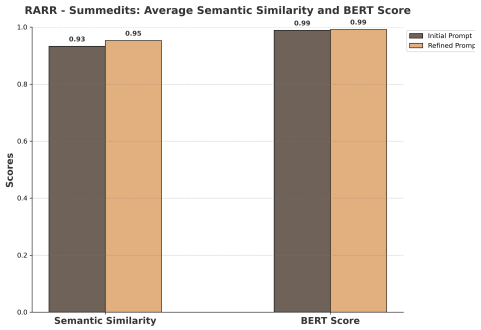
(b) G-Eval Scores for ExpertQA

Figure 5.38: Comparison of Zero-shot and Few-shot ExpertQA for NLI and G-Eval.

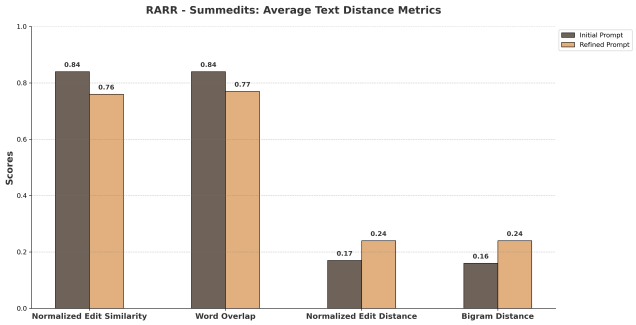
Understanding how few-shot examples work is crucial for fact-checkers and plays a critical role in how successful they are. However, enhanced few-shot version of ExpertQA shows no real improvement. This is because the dataset requires some flexibility on fact-checkers. Generic short few-shot examples are less helpful (or need tailored, highly specific examples — which were not explored due to time constraints). That's why COVE zero-shot, without any modifications, performs better than RARR already in this dataset. RARR zero-shot version also does not yield better results than COVE, and this explains some gaps such as less freedom in RARR work for handling long-texts.

5.2.3 Focused Context and Specificity

In the following experiments comparing initial prompt and refined prompt setups, for search tool Bing Search (snippets) is used, and dataset used is SummEdits.

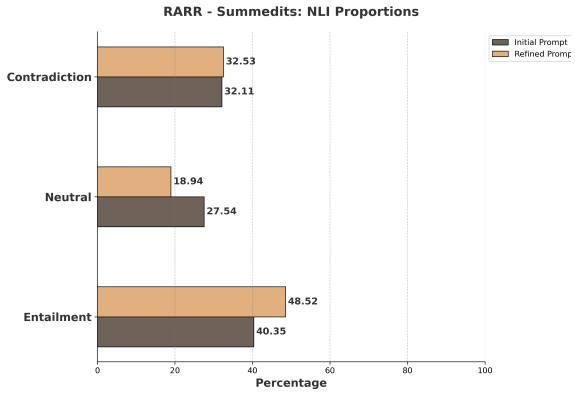


(a) Semantic BERT Scores for Summedits

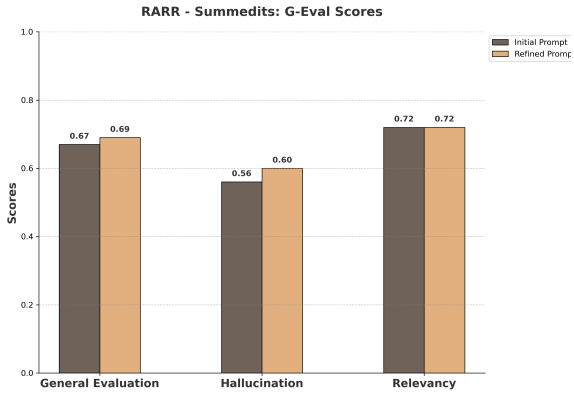


(b) Text Distances for Summedits

Figure 5.39: Comparison of Initial and Refined Prompts, Summedits for Semantic Similarity and Text Distances.



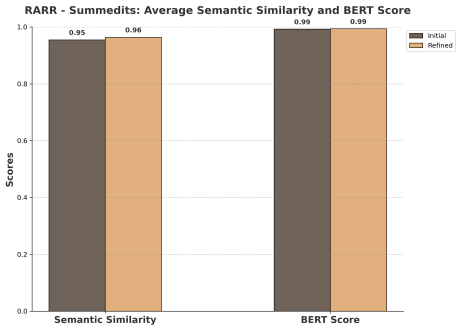
(a) NLI Proportions for Summedits



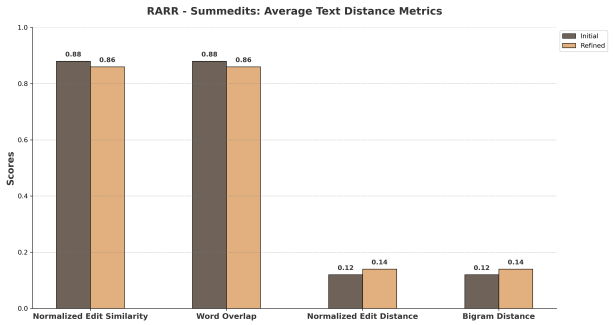
(b) G-Eval Scores for Summedits

Figure 5.40: Comparison of Initial and Refined Prompts, Summedits for NLI and G-Eval.

In this part, the setup is the same with previous setting, except the search engine switch from Bing to Google.

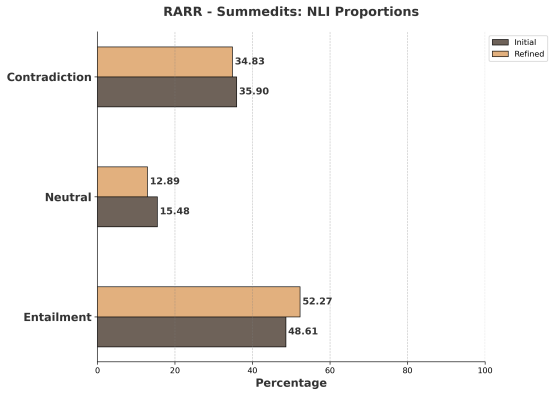


(a) Semantic BERT Scores for Summedits

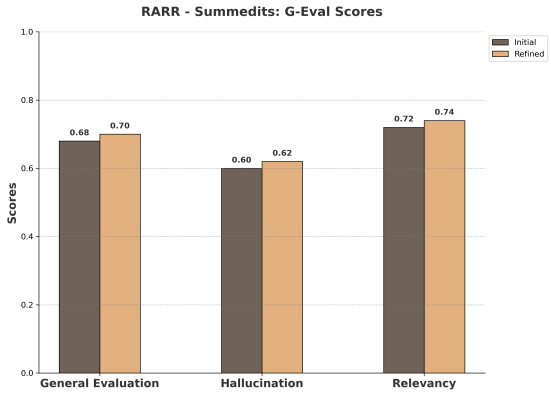


(b) Text Distances for Summedits

Figure 5.41: Comparison of Initial and Refined Prompts, Summedits for Semantic Similarity and Text Distances.



(a) NLI Proportions for Summedits



(b) G-Eval Scores for Summedits

Figure 5.42: Comparison of Initial and Refined Prompts, Summedits for NLI and G-Eval.

The main change is refining the prompts to be more specific. Each prompt now addresses the issues found in pipeline after deep diving generated results and checking them one by one. The key changes are:

- *Decontextualization Improvements*: Refined prompts to fix the issue of some questions being too general and lacking context. For example, instead of asking, “Who died 32 years ago?”, now asking, “Did Bourne Randolph die 32 years ago?”
- *Clarity/Detail*: Providing extra context when needed. For instance, instead of asking, "Where was Stanford Prison Experiment conducted?", now asking "Was the Stanford Prison Experiment conducted in the basement of Encina Hall at Stanford University?"
- *Spelling/Grammar*: Fixed minor spelling and grammar errors

These updates led to a good achievement: scoring over 0.5 in entailment and above 0.7 in general evaluation—setting a new record on SummEdits, excluding the first results by adding the original document itself.

5.3 Distribution of Hallucination Types

For this analysis, a subset of SummEdits samples that scored less than 0.5 in factuality was used. These samples were then passed to GPT-4o, from OpenAI, to identify hallucination types. Below is the prompt used:

```
"""
1) Given the following original text and corrected text, identify the
   types of hallucinations present in the corrected text according to
   the taxonomy provided.
2) Focus on identifying the deepest (leaf) hallucination types.
3) Terms like 'Factual Hallucinations' or 'Verifiable Falsehoods' are
   not specific enough and inner nodes should not be returned. 1, 1.1, and 2 should
   not be returned.

{taxonomy}

{one_shot_examples}

Original Text: "{original_text}"
Corrected Text: "{corrected_text}"
Hallucination Types (deepest level only):
"""
```

Table 5.6 shows the distribution of hallucinations, in the proposed categories and subcategories:

Type	Category	Subcategory	RARR (%)	COVE (%)
Factual Hallucinations	1.1 Verifiable Falsehoods	1.1.1 Numeric Hallucinations	24	15
		1.1.2 Temporal Hallucinations	14	12
		1.1.3 Location Hallucinations	19	21
		1.1.4 Reasoning Hallucinations	28	37
	1.2 Unverifiable Falsehoods	-	15	16

Table 5.6: Distribution of Hallucination Types in Terms of RARR and COVE Percentages

Factual Hallucinations

Factual hallucinations are hallucinations that involve inaccurate or fabricated information. The most dominant factual hallucinations are Reasoning Hallucinations, accounting for 28% in RARR and as high as 37% in COVE.

Numeric Hallucinations account for 24% of RARR. This demonstrates a recurrent problem with correct number handling. Then, there are two types of hallucinations: temporal and locational. They appear less often, but enough to indicate difficulties with dates or locations.

Faithfulness Hallucinations

When it comes to faithfulness hallucinations, there's a different situation. Instructional errors and self-inconsistencies are pretty rare, near zero in RARR and COVE. Faithfulness hallucinates does not apply to experiments in this dataset.

6 Discussion

This chapter summarizes the research findings and discusses the limitations of the study along with some proposals for future research.

6.1 Key Findings

This section reviews the methodologies and data sources used in the research, along with the limitations of language models (LLMs) in generating accurate information.

Which Fact-Checker Provides Better Corrections?

Comparative results reveal interesting insights and show that COVE is better for more expressive and courageous corrections with a zero-shot setup as shown in the LongFact dataset, while RARR optimizes for relevancy to the original text.

RARR also tends to perform more careful or simple corrections because it includes a simple yet effective rule: if the editor step in the pipeline makes an excessive change, this change is not allowed and is reverted. The distance between the original claim and the edited claim is measured by the Levenshtein distance. Examples of corrections are provided in Table 6.1.

Summary Type	Content
Gold Summary	The founder of the troubled digital start-up Ozy Media has been arrested on fraud charges as part of a scheme to prop up the financially struggling company.
Hallucinated Summary (input to fact-checker)	The founder of the **triumphant** digital start-up Ozy Media has been arrested for fraud charges as part of a scheme to **sell** the financially **thriving** company Ozy Media.
RARR (Bing snip.)	The founder of the **ailing** digital start-up Ozy Media has been arrested on fraud charges as part of a plan to **support** the financially **struggling** company.
CoVe (Bing snip.)	The CEO of Ozy Media, Carlos Watson, has been arrested on fraud charges. Ozy Media is facing financial **struggles** and there is a scheme in place to **prop up** the company. However, there is no information or news updates regarding the CEO's arrest. The current status of Ozy Media is that it has shut down, and its CEO has been accused of deceiving potential investors. The company's chairman, Marc Lasry, has stepped down.

Table 6.1: Example of an instance from SummEdits with final outputs of CoVe and RARR, both based on Bing search snippets. The RARR correction was more concise, while CoVe was more informative and detailed.

Selection of "News" Articles in SummEdits

The decision to emphasize "news" articles in the SummEdits dataset was driven by their structured format. News articles are easier to retrieve and fact-check using search engines. Other article categories in the SummEdits dataset (bill summaries, e-commerce summaries, or scientific TL; DRs) often lack context or key identifiers. For example, an e-commerce

summary might state, *"The company is increasing its dividend to $0.15 per share"* but fail to name the company, making it almost impossible to fact-check. In contrast, news articles often offer comprehensive context. For instance, the article specifies, *"A report by Forbes found that Binance secretly transferred $1.8 billion worth of funds to several hedge funds in August 2022."*, enabling accurate retrieval and validation. This makes news articles particularly suitable for these experiments.

Do Snippets Perform Better Than Full-Text Retrieval?

Passages do provide more context, but snippets are better at delivering spot-on information, even though they are shorter. This simplicity helps large language models (LLMs) catch and fix mistakes more easily. Snippets provide approximately an 8% factuality boost compared to passages for fact-checking. Additionally, search engines are making snippets longer and more detailed, which makes them even more suitable for LLMs.

Below Table 6.2, shows an example of retrieving chunks from full-text websites is suffering from numerical hallucination:

Summary Type	Content
Gold Summary	Bird flu was not deemed to be a threat to humans until 1997, after an outbreak in Hong Kong. Since then, around 870 infections have been reported worldwide, with 457 deaths in **21 countries**.
Scrape page + ranking	Since the bird flu outbreak in Hong Kong in 1997, more than **890** sporadic human infections with A(H5N1) bird flu viruses have been reported worldwide, resulting in 457 deaths in **23 countries**.
Search Snippets	Bird flu was not regarded as a threat to humans until 1997, after an outbreak in Hong Kong. Since then, around 870 infections have been reported worldwide, with 457 deaths in **21 countries**.

Table 6.2: Example of an instance from SummEdits with final outputs of RARR, one based on Google search snippets, the other based on full-page retrieval. Full-page setup retrieving topK chunks.

Fact-checking challenges with Internal LLM Knowledge

One of the key challenges identified in this research is the inherent limitation of LLMs when relying solely on internal knowledge. While models such as GPT-4 can generate factually relevant content, they are often restricted by their training data, which may become outdated or lack necessary details. For example, when the LLM was queried about *"Quantum Field Theory"*, it provided a broad overview, but its response lacked the latest insights from current research, such as new findings on *spontaneous symmetry breaking* and *virtual particles*. Moreover, models without access to real-time information frequently return incomplete or partially correct answers, such as stating, *"I'm sorry, as an AI language model, I don't have access to real-time information."*

Is Wikipedia a Reliable Knowledge Source?

During LongFact experiments, evaluations are realized by FActScore. Running FActScore requires a reliable knowledge source. However, Wikipedia as a knowledge source for fact-checking has the following main problems:

1. **Biases in Content**: Wikipedia is known to have various biases [57], [58], including political [59], persistent [60], gender [61], cultural [62], and framing bias [63].
2. **Coverage and Notability Issues**: Some articles in Wikipedia can be incomplete or lacking depth, creating a skewed representation of knowledge. Generated long-form texts contain more atomic facts than a simple one-sentence claim, and, therefore, require more comprehensive knowledge source(s).
3. **Vulnerability to False Information and Vandalism**: Wikipedia allows open editing, and articles can contain false information at any point in time [64].

Despite its widespread use, these limitations should be considered while designing fact-checking systems. In cases where there are no ground truth answers available, LM generated answers was evaluated using FActScore. However, fact-checkers may fail to capture all the necessary information to detect and correct hallucinations since all of the aspects of generated answers are not covered on respective Wikipedia page(s).

Is DuckDuckGo Able to Return Relevant Search Results?

The evaluation of DuckDuckGo's search capabilities revealed mixed results. While the search result links themselves were often accurate and relevant, the snippets provided by DuckDuckGo were less useful. They tended to repeat information rather than provide additional context, limiting their utility in long-form fact-checking tasks.

6.2 Limitations and Future Work

This study gives useful insights into fact-checking methods for Large Language Models (LLMs) by running different experiment setups. However, it has some limitations. One key issue is that it focuses only on specific fact-checkers, namely COVE and RARR [35], [36]. By looking at only these two, research might have missed other important aspects of fact-checkers.

Another limitation is the use of only a subset of the ExpertQA dataset. Due to high computational costs, the full dataset was not used.

Some ideas that could improve fact-checkers were not explored due to time constraints. Integrating these applications in industrial use cases would require more effort on optimization side since these pipelines are quite inefficient. A single fact-checking along with the evaluation might take up to 2 minutes, depending on the LLM and method used. Cost and performance wise, these models have much room for improvement. Simple future directions for optimization might be merging multiple LLM calls into one single LLM call, caching, merging prompts, etc.

Firstly, better connecting steps in the fact-checking pipeline could help. For example, if the search tool fails to find good evidence, question generator could be called again, to create shorter or simpler questions. This kind of back-and-forth could improve the retrieval of relevant information.

Secondly, creating a prompt adjustment step, such as a few-shot example generator, is also another possible improvement. Having prompts tailored to problem or dataset before checking claims can be beneficial. Adjustments should consider factors such as topic, style, tone, etc. for prompts.

Thirdly, improving routers within COVE is also promising. Right now, COVE router uses following prompts: `WIKI_CHAIN`, `MULTI_CHAIN`, and `LONG_CHAIN`:

> `WIKI_CHAIN`: For questions looking for a list or set of entities.
>
> `MULTI_CHAIN`: For questions with multiple parts, each needing separate answers from different text spans.
>
> `LONG_CHAIN`: For questions that need long answers.

Refining prompts, especially for `LONG_CHAIN`, could improve the process. There are also overlaps between categories like `MULTI_CHAIN` and `LONG_CHAIN`, which can be confusing for language models. Fixing these overlaps might improve accuracy.

Using more than one knowledge source is another potential improvement. Every method has its limits, so combining different sources could cover their weaknesses. Building a knowledge base specific to the problem area and using search engines to verify information could make fact-checking more reliable.

Finally, advanced search techniques (e.g. Google Dorks) could help find more relevant information. For example, using site-specific searches or keywords can guide searches to trusted sources:

> Query Example:
>
> `bauarbeiten 141 site:.de`
>
> This query looks for construction work (*bauarbeiten*) related to bus line 141 on the official MVV Munich website.

An example snippet:

> *"Bus 141 / 170 - Anton-Will-Straße > Rockefellerstraße: Diversion due to construction work on Neuherbergstraße (July 29th - October 25th, 2024) more."*

Using advanced search techniques, on-the-fly search queries that are customized for each atomic question can be created using LLMs. One atomic question might require checking resources published in 2023 (i.e. `after:2023-01-01 before:2023-12-31`), while another one needs to only look for official government websites (i.e. `inurl: gov`). This query generator can be added to pipelines. It should be enough to provide a list of search operators (e.g. `site:`, `inurl:`, `intext:`, `before`, `filetype:`, etc.)

Future work should explore these potential improvements. Combining token probabilities with Retrieval Augmented Generation (RAG) approaches, better connecting steps in pipeline, adding a few-shot prompt generator, improving the routing system, using multiple sources, and using advanced search methods are promising directions. Research in these areas could make fact-checking for LLM-generated content more accurate and reliable.

7 Conclusion

In this study, impact of many different setups on the performance of two state-of-the-art systems, COVE and RARR, for post-hoc hallucination correction is explored. Furthermore, improvements on post-hoc correction pipelines have been demonstrated on used datasets. Finally, a taxonomy of hallucination types and their distribution is examined.

Referring back to the research questions:

RQ1: What is an appropriate taxonomy for categorizing hallucinations in Large Language Models (LLMs)?
In recent literature on LLM factuality, hallucinations are generally classified into two main types. Proposed taxonomy also has factual and faithfulness hallucinations in the first level [11]. Expanding on this initial classification, a detailed taxonomy for hallucinations in LLMs is proposed as follows:

1. **Factual Hallucinations**
   - **Verifiable Falsehoods**:
     - **Numeric Hallucinations**: Mistakes about numbers (e.g. LLM saying a city has fewer people than it actually has).
     - **Temporal Hallucinations**: Errors with dates or times (e.g. LLM claiming an event happened in the wrong year).
     - **Location Hallucinations**: Misinformations on geographical parts or places (e.g. LLM writing Eiffel Tower is in Berlin).
     - **Reasoning Hallucinations**: Flawed logic (e.g. LLM concluding it must have rained just because the ground is wet).
   - **Unverifiable Falsehoods**: Claims that can not be proven right or wrong (e.g. imaginary situations, personal claims, hypothetical scenarios, or events that lack any form of proof or validation).
2. **Faithfulness Hallucinations**: This type looks at how well the model sticks to instructions and maintains consistency:
   - **Instructional**: When the model fails to follow user prompts (e.g. LLM giving a pizza recipe when asked for pasta).
   - **Self-inconsistency**: When the model contradicts itself (e.g. LLM saying hamburger is healthy in earlier messages while also accepting fast-foods are not healthy later messages in multi-turn conversations).

- **Contextual**: When the output does not match the context (e.g. LLM simply refutes or ignores the given context).

This taxonomy gives us a clearer picture of hallucinations in LLMs. Main contribution of this taxonomy is in the detailed subcategories. By breaking down the different types (e.g. numeric, location, etc.), it is possible to analyze further fact-checker behaviours.
This breakdown also helps in identifying common errors. To give an example, it might lead to noticing numerical hallucinations are overlooked by a fact-checker and accordingly adjusting specific prompts. One can tweak only the tools relevant to numerical hallucination detection and corrections. One advantage is that running taxonomy analysis is very straightforward since it is an LLM-based hierarchical classification realized by few-shot prompts. Any LLM can extract taxonomy from custom datasets, as long as there is ground truth text for comparison. Proposed approach also works across various domains.

RQ2: How can the Retrieval Augmented Generation (RAG) technique handle long contexts efficiently?

Handling long-text itself is a challenging problem. LLMs still suffer from the known issue 'Lost in the Middle' [65].
RAG can handle long contexts effectively in fact-checking by:

- **Using Atomic Parts**: Breaking down given queries into atomic claims and generating separate verification questions.
- **Using Search Snippets**: Pulling only small relevant information to help LLM reason better in search engine setup.
- **Custom Knowledge Bases**: Using domain-specific data to minimize knowledge-source limitations.
- **Optimizing Connections**: Connect or unify each step to avoid performance lags.
- **Search Operators**: Using specific search techniques to improve accuracy in retrieval step, or hybrid search

These steps help RAG stay efficient with long text inputs.

RQ3: How can we balance editing and faithfulness when improving generated text?
Hallucination correction while keeping it faithful to the original meaning is a challenging problem, and exists in recent fact-checking pipelines. One of the conclusions is that there is a trade-off between refinement and faithfulness. Experiments have also shown that COVE successfully corrects hallucinated inputs if correct evidence is retrieved — yet applies excessive refinements. On the other hand, RARR always generates more faithful output to the original summary — but fails to refine even if evidence is provided in some scenarios.
To balance editing and faithfulness:

- **Addıng Few-Shot Examples**: Provide the model with examples similar to desired ones for better output.
- **Keeping Relevant**: Avoid excessive refinement via text distance based rules.

In short, there should be a sweet spot in between refinements and relevancy. If a pipeline is designed to refuse excessive refinements, then have the correct evidence but no edit will be applied to hallucinated input. If a pipeline is designed to apply corrections aggressively, then it might lead to off-topic text generation.
While this thesis focus is limited to COVE and RARR, analysis and improvements aforementioned should be applicable and benefit existing and emerging fact-checking pipelines.

List of Figures

List of Tables

Bibliography

[1] C. Lyu, Z. Du, J. Xu, Y. Duan, M. Wu, T. Lynn, A. F. Aji, D. F. Wong, and L. Wang. "A Paradigm Shift: The Future of Machine Translation Lies with Large Language Models". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Ed. by N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue. Torino, Italia: ELRA and ICCL, May 2024, pp. 1339–1352. URL: https://aclanthology.org/2024.lrec-main.120.

[2] R. Patil and V. Gudivada. "A Review of Current Trends, Techniques, and Challenges in Large Language Models (LLMs)". In: *Applied Sciences* 14.5 (2024). ISSN: 2076-3417. DOI: 10.3390/app14052074. URL: https://www.mdpi.com/2076-3417/14/5/2074.

[3] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, and G. Wang. "Text Classification via Large Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by H. Bouamor, J. Pino, and K. Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 8990–9005. DOI: 10.18653/v1/2023.findings-emnlp.603. URL: https://aclanthology.org/2023.findings-emnlp.603.

[4] Z. Wang, Y. Pang, and Y. Lin. *Smart Expert System: Large Language Models as Text Classifiers*. 2024. arXiv: 2405.10523 [cs.CL]. URL: https://arxiv.org/abs/2405.10523.

[5] F. Wei, R. Keeling, N. Huber-Fliflet, J. Zhang, A. Dabrowski, J. Yang, Q. Mao, and H. Qin. "Empirical Study of LLM Fine-Tuning for Text Classification in Legal Document Review". In: *2023 IEEE International Conference on Big Data (BigData)*. 2023, pp. 2786–2792. DOI: 10.1109/BigData59044.2023.10386911.

[6] W. Zhu, H. Liu, Q. Dong, J. Xu, S. Huang, L. Kong, J. Chen, and L. Li. *Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis*. 2024. arXiv: 2304.04675 [cs.CL]. URL: https://arxiv.org/abs/2304.04675.

[7] D. M. Alves, J. Pombal, N. M. Guerreiro, P. H. Martins, J. Alves, A. Farajian, B. Peters, R. Rei, P. Fernandes, S. Agrawal, P. Colombo, J. G. C. de Souza, and A. F. T. Martins. *Tower: An Open Multilingual Large Language Model for Translation-Related Tasks*. 2024. arXiv: 2402.17733 [cs.CL]. URL: https://arxiv.org/abs/2402.17733.

[8] L. Basyal and M. Sanghvi. *Text Summarization Using Large Language Models: A Comparative Study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT Models*. 2023. arXiv: 2310.10449 [cs.CL]. URL: https://arxiv.org/abs/2310.10449.

[9] Y. Liu, K. Shi, K. S. He, L. Ye, A. R. Fabbri, P. Liu, D. Radev, and A. Cohan. *On Learning to Summarize with Large Language Models as References*. 2024. arXiv: 2305.14239 [cs.CL]. URL: https://arxiv.org/abs/2305.14239.

[10] W. Liang, Y. Zhang, Z. Wu, H. Lepp, W. Ji, X. Zhao, H. Cao, S. Liu, S. He, Z. Huang, D. Yang, C. Potts, C. D. Manning, and J. Y. Zou. *Mapping the Increasing Use of LLMs in Scientific Papers*. 2024. arXiv: 2404.01268 [cs.CL]. URL: https://arxiv.org/abs/2404.01268.

[11] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*. 2023. arXiv: 2311.052a32 [cs.CL]. URL: https://arxiv.org/abs/2311.05232.

[12] C. O'Connor and J. O. Weatherall. *The Misinformation Age: How False Beliefs Spread*. New Haven, CT, USA: Yale University Press, 2019.

[13] Z. Xu, S. Jain, and M. Kankanhalli. *Hallucination is Inevitable: An Innate Limitation of Large Language Models*. 2024. arXiv: 2401.11817 [cs.CL]. URL: https://arxiv.org/abs/2401.11817.

[14] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, and S. Shi. *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models*. 2023. arXiv: 2309.01219 [cs.CL]. URL: https://arxiv.org/abs/2309.01219.

[15] R. Ali. "Fiona Macpherson and Dimitris Platchias (Eds.), Hallucination: Philosophy and Psychology". In: *Phenomenology and the Cognitive Sciences* 15 (Jan. 2015). DOI: 10.1007/s11097-015-9413-3.

[16] J. D. Blom. *A Dictionary of Hallucinations*. Jan. 2010. ISBN: 978-1-4419-1222-0. DOI: 10.1007/978-1-4419-1223-7.

[17] S. Baker and T. Kanade. "Hallucinating faces". In: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)* (2000), pp. 83–88. URL: https://api.semanticscholar.org/CorpusID:6076353.

[18] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. "Survey of Hallucination in Natural Language Generation". In: *ACM Computing Surveys* 55.12 (Mar. 2023), pp. 1–38. ISSN: 1557-7341. DOI: 10.1145/3571730. URL: http://dx.doi.org/10.1145/3571730.

[19] A. F. Biten, L. Gomez, and D. Karatzas. *Let there be a clock on the beach: Reducing Object Hallucination in Image Captioning*. 2021. arXiv: 2110.01705 [cs.CV]. URL: https://arxiv.org/abs/2110.01705.

[20] A. Rohrbach, L. A. Hendricks, K. Burns, T. Darrell, and K. Saenko. *Object Hallucination in Image Captioning*. 2019. arXiv: 1809.02156 [cs.CL]. URL: https://arxiv.org/abs/1809.02156.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[23] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL]. URL: https://arxiv.org/abs/1910.13461.

[24] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[25] P. Koehn and R. Knowles. "Six Challenges for Neural Machine Translation". In: *Proceedings of the First Workshop on Neural Machine Translation*. Ed. by T. Luong, A. Birch, G. Neubig, and A. Finch. Vancouver: Association for Computational Linguistics, Aug. 2017, pp. 28–39. DOI: 10.18653/v1/W17-3204. URL: https://aclanthology.org/W17-3204.

[26] V. Raunak, A. Menezes, and M. Junczys-Dowmunt. *The Curious Case of Hallucinations in Neural Machine Translation*. 2021. arXiv: 2104.06683 [cs.CL]. URL: https://arxiv.org/abs/2104.06683.

[27] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. *On Faithfulness and Factuality in Abstractive Summarization*. 2020. arXiv: 2005.00661 [cs.CL]. URL: https://arxiv.org/abs/2005.00661.

[28] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. "On Faithfulness and Factuality in Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 1906–1919. DOI: 10.18653/v1/2020.acl-main.173. URL: https://aclanthology.org/2020.acl-main.173.

[29] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi. *FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation*. 2023. arXiv: 2305.14251 [cs.CL]. URL: https://arxiv.org/abs/2305.14251.

[30] J. Kim, S. Park, Y. Kwon, Y. Jo, J. Thorne, and E. Choi. *FactKG: Fact Verification via Reasoning on Knowledge Graphs*. 2023. arXiv: 2305.06590 [cs.CL]. URL: https://arxiv.org/abs/2305.06590.

[31] I.-C. Chern, S. Chern, S. Chen, W. Yuan, K. Feng, C. Zhou, J. He, G. Neubig, and P. Liu. *FacTool: Factuality Detection in Generative AI – A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios*. 2023. arXiv: 2307.13528 [cs.CL]. URL: https://arxiv.org/abs/2307.13528.

[32] S. Lin, J. Hilton, and O. Evans. *TruthfulQA: Measuring How Models Mimic Human Falsehoods*. 2022. arXiv: 2109.07958 [cs.CL]. URL: https://arxiv.org/abs/2109.07958.

[33] J. Kasai, K. Sakaguchi, Y. Takahashi, R. L. Bras, A. Asai, X. Yu, D. Radev, N. A. Smith, Y. Choi, and K. Inui. *RealTime QA: What's the Answer Right Now?* 2024. arXiv: 2207.13332 [cs.CL]. URL: https://arxiv.org/abs/2207.13332.

[34] J. Vladika and F. Matthes. *Scientific Fact-Checking: A Survey of Resources and Approaches*. 2023. arXiv: 2305.16859 [cs.CL]. URL: https://arxiv.org/abs/2305.16859.

[35] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston. *Chain-of-Verification Reduces Hallucination in Large Language Models*. 2023. arXiv: 2309.11495 [cs.CL]. URL: https://arxiv.org/abs/2309.11495.

[36] L. Gao, Z. Dai, P. Pasupat, A. Chen, A. T. Chaganty, Y. Fan, V. Y. Zhao, N. Lao, H. Lee, D.-C. Juan, and K. Guu. *RARR: Researching and Revising What Language Models Say, Using Language Models*. 2023. arXiv: 2210.08726 [cs.CL]. URL: https://arxiv.org/abs/2210.08726.

[37] O. Honovich, L. Choshen, R. Aharoni, E. Neeman, I. Szpektor, and O. Abend. "Q^2: Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7856–7870. DOI: 10.18653/v1/2021.emnlp-main.619. URL: https://aclanthology.org/2021.emnlp-main.619.

[38] J. Wei, C. Yang, X. Song, Y. Lu, N. Hu, J. Huang, D. Tran, D. Peng, R. Liu, D. Huang, C. Du, and Q. V. Le. *Long-form factuality in large language models*. 2024. arXiv: 2403.18802 [cs.CL]. URL: https://arxiv.org/abs/2403.18802.

[39] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. *BERTScore: Evaluating Text Generation with BERT*. 2020. arXiv: 1904.09675 [cs.CL]. URL: https://arxiv.org/abs/1904.09675.

[40] T. Gao, X. Yao, and D. Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *CoRR* abs/2104.08821 (2021). arXiv: 2104.08821. URL: https://arxiv.org/abs/2104.08821.

[41] V. I. Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics. Doklady* 10 (1965), pp. 707–710. URL: https://api.semanticscholar.org/CorpusID:60827152.

[42] L. Yujian and L. Bo. "A Normalized Levenshtein Distance Metric". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 1091–1095. DOI: 10.1109/TPAMI.2007.1078.

[43] C.-Y. Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[44] D. Z. Korman, E. Mack, J. Jett, and A. H. Renear. "Defining Textual Entailment". In: *Journal of the Association for Information Science and Technology* 69 (2018), pp. 763–772.

[45] A. Mishra, D. Patel, A. Vijayakumar, X. L. Li, P. Kapanipathi, and K. Talamadupula. "Looking Beyond Sentence-Level Natural Language Inference for Question Answering and Text Summarization". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou. Online: Association for Computational Linguistics, June 2021, pp. 1322–1336. DOI: 10.18653/v1/2021.naacl-main.104. URL: https://aclanthology.org/2021.naacl-main.104.

[46] P. Laban, T. Schnabel, P. N. Bennett, and M. A. Hearst. "SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization". In: *Transactions of the Association for Computational Linguistics* 10 (Feb. 2022), pp. 163–177. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00453. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00453/1987014/tacl_a_00453.pdf. URL: https://doi.org/10.1162/tacl%5C_a%5C_00453.

[47] J. Steen, J. Opitz, A. Frank, and K. Markert. "With a Little Push, NLI Models can Robustly and Efficiently Predict Faithfulness". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by A. Rogers, J. Boyd-Graber, and N. Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 914–924. DOI: 10.18653/v1/2023.acl-short.79. URL: https://aclanthology.org/2023.acl-short.79.

[48] P. He, J. Gao, and W. Chen. *DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing*. 2023. arXiv: 2111.09543 [cs.CL]. URL: https://arxiv.org/abs/2111.09543.

[49] M. Laurer, W. van Atteveldt, A. Casas, and K. Welbers. *Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT-NLI*. Dec. 2022. URL: osf.io/wqc86.

[50] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu. *G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment*. 2023. arXiv: 2303.16634 [cs.CL]. URL: https://arxiv.org/abs/2303.16634.

[51] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. 2023. arXiv: 2306.05685 [cs.CL]. URL: https://arxiv.org/abs/2306.05685.

[52] C. AI. *deepeval*. https://github.com/confident-ai/deepeval. 2023.

[53] K. Krishna, E. Bransom, B. Kuehl, M. Iyyer, P. Dasigi, A. Cohan, and K. Lo. "LongEval: Guidelines for Human Evaluation of Faithfulness in Long-form Summarization". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Ed. by A. Vlachos and I. Augenstein. Dubrovnik, Croatia: Association for

Computational Linguistics, May 2023, pp. 1650–1669. DOI: 10.18653/v1/2023.eacl-main.121. URL: https://aclanthology.org/2023.eacl-main.121.

[54] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi. *FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation – Code*. https://github.com/shmsw25/FActScore. Accessed: 2024-06-05. 2023.

[55] L. Gao, Z. Dai, P. Pasupat, A. Chen, A. T. Chaganty, Y. Fan, V. Zhao, N. Lao, H. Lee, D.-C. Juan, and K. Guu. *RARR: Researching and Revising What Language Models Say, Using Language Models*. https://github.com/anthonywchen/RARR. Accessed: 2024-06-05. 2023.

[56] Ritun16. *Chain of Verification – Unofficial GitHub Implementation*. https://github.com/ritun16/chain-of-verification. Accessed: 2024-06-05. 2023.

[57] S. Greenstein and F. Zhu. "Is Wikipedia Biased?" In: *American Economic Review* 102.3 (May 2012), pp. 343–48. DOI: 10.1257/aer.102.3.343. URL: https://www.aeaweb.org/articles?id=10.1257/aer.102.3.343.

[58] J. Koerner. "Wikipedia has a bias problem". In: (2019).

[59] C. Hube and B. Fetahu. "Detecting Biased Statements in Wikipedia". In: Apr. 2018, pp. 1779–1786. DOI: 10.1145/3184558.3191640.

[60] B. Martin. "Persistent bias on Wikipedia: Methods and responses". In: *Social Science Computer Review* 36.3 (2018), pp. 379–388.

[61] J. Reagle and L. Rhue. "Gender bias in Wikipedia and Britannica". In: *International Journal of Communication* 5 (2011), p. 21.

[62] E. S. Callahan and S. C. Herring. "Cultural bias in Wikipedia content on famous persons". In: *Journal of the American society for information science and technology* 62.10 (2011), pp. 1899–1915.

[63] G. S. Andre Oboler and R. Stern. "The Framing of Political NGOs in Wikipedia through Criticism Elimination". In: *Journal of Information Technology & Politics* 7.4 (2010), pp. 284–299. DOI: 10.1080/19331680903577822. eprint: https://doi.org/10.1080/19331680903577822. URL: https://doi.org/10.1080/19331680903577822.

[64] J. Tramullas, P. Garrido-Picazo, and A. I. Sánchez-Casabón. "Research on Wikipedia Vandalism: a brief literature review". In: *Proceedings of the 4th Spanish Conference on Information Retrieval*. 2016, pp. 1–4.

[65] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. *Lost in the Middle: How Language Models Use Long Contexts*. 2023. arXiv: 2307.03172 [cs.CL]. URL: https://arxiv.org/abs/2307.03172.