Imperial College London

Department of Computing

Verifiable Text Generation

Author: Zeeshan Ahmed

Project Supervisor: Dr. Thomas Lancaster

December 1, 2025

Abstract

Large Language Models (LLMs) are increasingly used in professional and educational settings, yet their tendency to hallucinate makes verification essential. Attribution - the inclusion of supporting sources alongside generated text - offers a direct mechanism for grounding model outputs. This project systematically evaluates three attribution methods for Attributed Question Answering (AQA): Post-Retrieval Answering, Post-Generation Attribution via re-prompting (with short/long instructions), and Post-Generation Attribution via TF-IDF searching. Experiments were conducted across multiple QA datasets covering short-form and long-form with evaluation focusing on both answer correctness and citation quality.

Our findings highlight how attribution methods behave differently depending on task type and evaluation setting, with post-hoc methods offering flexibility but introducing variability - whereas post-retrieval proves to be robust and effective in all datasets. By presenting a comparative analysis across multiple dimensions, this work contributes to a clearer understanding of the strengths, weaknesses, and trade-offs involved in designing verifiable text generation.

Contents

Chapter 1

Introduction

Generative AI for text has been rapidly developing, and services like ChatGPT and Gemini have become part of the workflow for many professionals, students and teachers alike. AI text generation is powered by Large Language Models (LLMs), which are prone to hallucinating - generating plausible but false statements. This makes verification essential, as users may inadvertently accept misleading or dangerous information - a phenomenon that can mislead even experienced professionals [1].

The term "verifiable" entails that the text generation should be correct and grounded. Various techniques improve the correctness of LLM outputs, such as Retrieval-Augmented Generation (RAG) [2], Chain-of-thought reasoning [3] and enhanced pre-training strategies [4]. The field of LLM attribution, where developers and users can see the source of an answer, is the most direct way to verify text generation and make sure it is grounded. This field has started to gain traction by researchers and is discussed in this paper.

Popular commercial generative search engines have started including in-line citations in their generated text. However, studies show that these attributions often remain unreliable. In early 2025, audits of eight systems revealed that over 60% of citations were incorrect [5]. These engines rely on black-box LLMs and proprietary search backends, therefore it's difficult to trace where the failure occurs - whether during retrieval, generation, or citation matching. Researchers have begun analysing how different stages in an end-to-end pipeline affect citation quality [6].

There are three main techniques to LLM attribution [7]: **Direct Generated Attribution**, **Post-Retrieval Answering** and **Post-Generation Attribution**.

Direct generated attribution is based on the model's parametric knowledge only. Asking models to self-attribute often improves the truthfulness of their output [8] but because it is limited to pre-training data, this approach struggles when answering domain-specific knowledge-based questions and also when citing references for them.

Post-retrieval answering includes the relevant, out-of-model knowledge into the original prompt/query by using retrievers for web pages or documents. This approach is used in RAG-style pipelines and significantly enhances the attribution accuracy of LLMs.

Post-generation attribution is a method of attributing claims with references after generation via retrievers or further prompting. This approach allows the LLM to keep its original readability and creativeness, but still allowing the user to check and verify the references. RARR [9] combines this with post-editing to remove unsupported claims made in the original generation.

Combining these techniques into a unified architecture is known as an **attribution system**. Notable early systems include WebGPT [10], SPARROW [11] and GopherCite [12] - incorporating retrieval, citation-aware prompting, and confidence-based refusals; more recent systems include TRACE [13] and ABE [14]. These designs laid the groundwork for the next generation of AI-powered tools, like Deep Research [15] agents and the like.

1.1 Motivation and Aims

Large language models are now commonly used in settings demanding both factual correctness and credible evidence. Attribution enhances trust, but often at a cost: answer quality, completeness, or conciseness can suffer under stronger attribution constraints. In this work, we compare attribution methods across multiple datasets and domains, evaluating correctness and citation quality side-by-side. Our aim is to identify which methods yield strong, reliable attribution without compromising answer quality - across short-form and long-form, list-style and explanation-style answers.

More formally, we systematically evaluate methods for *attributed question answering* (AQA). Generating statements $\mathcal{S} = \{s_i\}$ with supporting citations $\mathcal{C}_i \subset \mathcal{D}$, across a diverse range of QA datasets. We focus on how different prompting strategies and post-hoc attribution mechanisms affect both the correctness of model outputs and quality of the citations, also investigating their trade-offs.

These aims translate into two guiding research questions: (1) Which attribution method and LLM best balances answer correctness and citation quality? and (2) When citations are attached post-hoc, how close can performance come to prompt-integrated attribution?

Chapter 2

Background

This section will expand on the different aspects involved in LLM attribution: namely the **sources**, **datasets**, **approaches** and **evaluation**. An understanding of the transformer LLM architecture [16], which underpins modern language models, is assumed.

2.1 Pre-training Data and Out-of-knowledge Data

The two categories of sources of where a LLM would get its information for generation is the model's parametric knowledge or out-of-model knowledge.

LLMs are trained on by incredible amounts of data from the web. Depending on the data, objectives, and methods employed when training, LLMs are able to perform different tasks and have particular behaviours. Modern systems can now be prompted and fine-tuned to generate citations from pre-training data. Because verifiability has become much more of a concern in generated text [17], research in training objectives for grounding is developing [18].

Connecting LLMs to out-of-model knowledge, however, is the most common and obvious strategy for verifiable generation. This is usually referred to as Retrieval-Augmented-Generation (RAG). RAG equips the model with a retriever that can search a text corpus or the web for relevant documents at inference time, which the model then concatenates with the query on to generate an answer. This grounds the LLM's output in up-to-date and relevant evidence. RAG does not necessarily mean an answer is verifiable, as it still needs to be coupled with attribution logic. By providing retrieved sources and evidence in the prompt itself, then instructing the model to generate citations for claims, the text generation becomes verifiable.

2.2 Datasets

There are several common datasets for LLM attribution. They play multiple roles as they can be used for training, fine-tuning or evaluating. For attribution, **question-answering** and **text summarisation** are the main tasks the datasets are curated for and are typically open-domain. Attribution granularity can range from answer-level citation (coarse-grained) or sentence/claim-level citation (fine-grained). It is important to note that many of the datasets used specifically for citation tasks are extensions of existing datasets, like ELI5 [19], TruthfulQA [20] and TriviaQA [21]. They build upon, modify or unify to include citations, indexing conventions, or standardised input-output structures.

OpenAI's **WebGPT** project gathered a dataset of human-written answers and preference comparisons (of different answers for the same query) that include references of relevant webpages. The data was used to fine-tune and evaluate WebGPT on long-form answers with citations. It's open-domain, meaning there are multiple subject matters and the granularity is sentence-level.

CiteBench also provides a unified dataset for LLM attribution; it combines datasets ABURAED, CHEN, LU and XING which were all summarisation tasks - then adapted them for citation. CiteBench was the first unified benchmark for scientific citation text generation and its granularity is also sentence-level.

ALCE [22] is a major work for LLM citation benchmarks, but also provide open-domain QA data by selecting 1,000 examples from the development sets of **ASQA** [23] (A short-answer factoid QA dataset), **QAMPARI** [24] (listed entities dataset) and **ELI5** [19] (a long-form explanation dataset sourced from the Reddit forum "Explain Like I'm Five"). This results in a total of 3,000 examples. ALCE's construction is not meant for training citation generation as there is no supervised signal or training split provided. Instead, the focus is on automatic evaluation and benchmarking.

Many attribution datasets rely on human-annotated data by domain experts. Humans may cite particular web pages or documents for gold answers that the retriever of an attribution pipeline may not have access to but are still just as valid. By using model-based methods to label whether citations support a claim, manual annotations become less important and thus datasets are able to scale in their size - like AttrEval-Sim [25] (over 60k samples) and CAQA [26] (over 160k samples).

2.3 Approaches to Attribution

2.3.1 Direct Generation Attribution

Direct attribution methods aim to have the model produce citations from its parametric knowledge. Hallucination is common when LLMs are prompted for evidences when answering domain-specific knowledge-based questions [27, 17]. Sources that are generated are sometimes loosely related by keywords and biased towards corpuses like Wikipedia, due to their pre-training. For domains where authoritative sources are essential, like medicine and law, hallucinations pose high risks, and attribution from parametric knowledge helps reduce hallucination.

There are many prompt-based techniques that help models cite relevant sources from their parametric knowledge. According-to [28] explores **according-to [source]** prompting for citation. In their experiments, they tested (1) no appended prompts (2) appended prompts with "according-to [source] type phrases" and (3) anti-grounding prompts. In their results, they claimed "more quotations correlate with fewer hallucinations" and showed that according-to prompts improved the QUIP-Score (the percentage of the model's generation that exists as exact quotes in the pre-training corpus) and anti-grounding prompts reduced the QUIP-Score over multiple domains and corpora.

Feedback learning LLM researchers explored a **feedback learning loop** [29] to improve generated citations. Their proposed algorithm had 3 key steps after generation. (1) Evaluate the generated answer with a critic model, assigning scores for fluency, correctness and citation. (2) Provide feedback to the LLM based on the aspect scores. (3) Create a refined answer based on the feedback. The feedback iteration process was evaluated against the base ChatGPT model and outperformed it on all metrics. The feedback algorithm can easily be incorporated to all approaches to attribution.

2.3.2 Post-Retrieval Answering

In post-retrieval answering, there is an explicit retrieval component that fetches documents before or during generation; and the LLM's job is to answer from those documents and quote relevant passages, ideally without introducing outside information. This is often referred to as **RAG** (Retrieval-Augmented Generation). When considering attribution in RAG systems, citation recall and precision is highly dependent on the retriever, but consistently shows a large improvement compared to direct generation attribution.

The basic idea is that based on the user's query: a retriever fetches related documents, appends or prepends the entire document or snippets from the document (depending on context window length) to the original query, then is finally prompted to answer the query based on the retrieved evidences with citation instructions. Because the evidences are in the prompt itself, LLMs are able to be more grounded and transparent with their generation and citations, as opposed to relying solely on their pre-training data.

ALCE benchmark framework proposes several variants of post-retrieval prompting. Its *Vanilla* method involves simply prompting the LLM with the top-k retrieved passages (where k is scaled by context window size) with a fixed instruction and in-context demonstration(s)), demanding citation for all factual claims. Vanilla achieves strong correctness and citation quality under their corresponding evaluation. To address the limitation of context window size, ALCE also explores a *Summ/Snippet* approach. In Summ/Snippet, retrieved passages are compressed: "summaries" are abstractive condensations, "snippets" are extracted spans. These compressed versions reduce token usage per passage, enabling more documents to be fed into the prompt. The paper finds that Summ/Snippet improves correctness (because more evidence is accessible) though citation quality worsens in some cases due to information loss during compression.

LLatrieval [30] proposes a framework which improves a retrieval pipeline for citation, "where the LLM iteratively provides feedback to the retrieval through verify-update iterations". Their pipeline works as follows: given a user query q, a retriever (BM25 used in the study) fetches D, the top-k documents. The LLM is prompted to evaluate whether $\mathcal{D}$ is sufficient to answer q via (1) a binary yes or no classification or (2) rating the documents 0-10. If the LLM flags the retrieval as insufficient, LLatrieval refines the retrieved documents by Progressive Selection (PS) and Missing-Info Query (MIQ). PS refines D by swapping out irrelevant ones and adding stronger candidates. MIQ uses the LLM to inspect D and identify missing or weakly covered subtopics of the query, generates a new query to target the missing information and is rerun through the retriever to get the additional documents. This process is then looped until the LLM confirms that the current pool of documents D satisfies the original query; generating a final answer with embedded, fine-grained citations to the documents. When evaluated on the ALCE benchmark [22]; significant improvements in citation F1 were recorded against basic RAG and static - causing them to become the bottleneck of the entire pipeline and limiting the overall performance.

SearChain [31] focuses on post-retrieval answering for "complex" tasks like multi-hop Q&A and long-form Q&A. The framework uses a LLM to decompose a complex question into a sequence of sub-queries with tentative answers. These initial question-answer pairs are nodes of what they refer to as a global reasoning chain called Chain-of-Query, which starts out as a linear chain. For each node, SearChain uses IR to fetch documents and verifies the LLM's tentative answers. A confidence score is allocated for the node using a reader model based on how reliable the reader believes the retrieved document supports or contradicts the model's answer to the sub-query. If the IR detects the node needs to be corrected or provided with knowledge, it gives feedback to the LLM and regenerates a new CoQ, converting the chain into a tree with branches where the new CoQ is the new branch. They refer to this process as node-identify Depth-first Search on Tree-of-Reasoning. Once the tree is fully verified and grounded, the system generates the final output, presenting an answer with citations to the source documents for each reasoning step. SearChain's method improves upon challenges faced in complex knowledge-intensive tasks. For example, the interaction between IR and the LLM in SearChain becomes a depth-first traversal strategy on a tree as opposed to a fixed sequence of reasoning steps, allowing a model to dynamically change the reasoning path based on evidence.

2.3.3 Post-Generation Attribution

The section before covered retrieval-then-generate methods, this section will cover generate-then-retrieve. This approach usually reduces problems like semantic mismatch, where the query might retrieve documents that are topically related, but don't contain the needed fact; causing the model to produce an answer that seems plausible given the documents but it actually unsupported; thus hallucinate.

A prominent example of post-generation attribution is the **RARR** [9] approach. It works as follows: given a generated answer, break it into atomic factual claims then for each claim, search the web to retrieve k pages as evidences. A query-document relevance model is used to choose the best evidence for each query. If a claim is not fully supported by the source (a separate agreement model is used for this evaluation), the LLM is prompted to revise that portion of the answer to agree with the retrieved source. The final answer is attached with an attribution report of at most 5 references. If more than 5 references were used, a subset that maximises coverage of the attributable points is used, utilising the same relevance model at retrieval. By actively identifying and correcting unsupported claims based on real-time retrieval, RARR substantially enhances the verifiability and trustworthiness of LLM-generated text, using a proactive approach to factual accuracy and source adherence.

A complementary approach in the post-generation attribution space is **CCVER** [32]. Once the LLM output is available, CCVER first performs claim decomposition, splitting a complex claim into yes/no sub-questions, via a language model (text-davinci-003 used in the study). Each sub-question is then issued as a query to a search engine under temporal constraints (restricting documents to those published before the claim date and excluding fact-check sites). Retrieved documents go through a fine-grained evidence retrieval stage, isolating relevant paragraphs. These are then synthesized into a claim-focused summary using a language-model-based summarisation component. CCVER pushes beyond simple binary support checks to a more nuanced, claim-aware post-generation verification process

2.3.4 Attribution Systems

Attribution systems combine multiple strategies into a pipeline. This is closer to what would be used commercially today, as companies would be able to pay the cost needed for hybrid methods and approaches to optimise performance in more parts of a pipeline than researchers.

GopherCite [12] and **WebGPT** [10] are two attribution systems that were released at similar times, and contributed to verifiable text generation greatly by exploring commercial search engines as the retriever for relevant documents, mitigating the issue of out-of-date documents for retrieval. WebGPT's approach mimics how a human researcher interacts with the web when answering a query and introduces a text-based web browser environment where the WebGPT model itself can "see" the current state of the browser, curating brief snippets as evidences for their final answer. GopherCite, on the other hand, utilises a larger context window with uncurated information from multiple pages - emphasising more on the reader comprehension ability of the model. Another key difference is that GopherCite's final answer is usually a single claim, and hence uses a single evidence for it, whereas WebGPT's final answer may have multiple claims and can have multiple citations.

The novelty in WebGPT is that it uses a text-based web-browsing environment for information retrieval, that a fine-tuned language model interacts with. The model is able to execute commands like "search <query>", "Clicked on link <link ID>" and "quote <text>". When the quote command is executed, the relevant extract is stored, along with the page title and domain name. As it continues browsing, the quote snippets and page metadata is being built as its internal context for the query. When the model decides that it no longer needs to browse (maximum length of references or maximum number of actions) the answer generation model in the pipeline is then prompted with the question and references and composes its final answer. The answer has an inline, numbered citation style for referencing. The snippets from the original passages used as references ranges from a sentence to multiple paragraphs.

At inference time, GopherCite's approach works by performing a retrieval based on the user's query, retrieving a large, static set of top-k documents ($k \leq 10$). This collection of documents forms the primary context for the language model as it has been trained to deal with large token corpuses. A generator model is used to produce candidate answers (samples) based on each document in a round-robin fashion. Samples are then scored with a reward model, taught from human preferences and the sample with the highest reward is presented to the user with the respective document title and quote it was based on as its reference.

LaMDA focuses on dialogue interaction rather than strictly factual QA. They utilise models designed for dialogue and set an objective that quality of output for multi-turn context is paramount, and hence fine-tuning and evaluation are catered towards it, more so than GopherCite and WebGPT. LaMDA goes for a post-generation attribution approach rather than GopherCite's and WebGPT's post-retrieval answering approach. They claim that doing so allows the fine-tuning process to not compromise in their safety and quality objectives. This is reflected in their respective outputs as GopherCite and WebGPT embed quotes directly into responses and use Reinforcement Learning techniques to optimise for this, whereas LaMDA responses have optional citations; trying to balance between conversational flow with factual correctness.

2.4 Evaluation

Evaluation for LLM attribution usually depends on the type of task; whether it is short-form or long-form question-answering. Researchers primarily use humans to assess model samples due to the lack of standardised and robust benchmarks for QA attribution tasks. Due to the costly and time-consuming nature of human assessment, research in automatic quantitative and categorisation evaluation for attribution is on the rise.

Below are typical metrics and benchmarks used when assessing verifiable text generation.

2.4.1 Metrics

For citation quality, the following metrics are key for evaluation.

- **Citation precision** is the proportion of cited sources that are actually relevant:

$$\text{Precision} = \frac{\text{Number of citations that entail the claim}}{\text{Total number of citations}}$$

- **Citation recall** measures coverage of necessary citations:

$$\text{Recall} = \frac{\text{Number of claims that are fully supported by their citations}}{\text{Total number of claims}}$$

- **F1 Score** is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

High precision means the model isn't citing extraneous or unrelated documents, high recall means the model is not leaving claims unsupported. For instance, if an answer makes 10 factual claims but only 7 have citations, recall is 70%. If it cites 5 sources but only 4 truly contain the information, precision is 80%.

Evaluation frameworks that utilise and build on the above metrics are:

- **Auto-AIS** [9]: citation precision and recall decided using an NLI model to determine if cited passages entail the claims in the response.
- **AttrScore** [25]: classifies citations in categories: attributable, extrapolatory or contradictory, then computes an overall F1 score based on their proportions. Classification is by an LLM, rather than an NLI model.
- **LQAC** [33]: extends AutoAIS to include partial supported claims in precision metric, using GPT-4o as the NLI model.
- **CiteEval-Auto** [34]: calculates a score that considers entire retrieval context, not just the cited passages. Like AttrScore and LQAC, an LLM is used for entailment classification.

Correctness is also important when assessing verifiable text generation, however it is a broad term when it comes to LLM attribution. This is because a generated text may be correct, but cites incorrect information - and vice versa. The general metrics used for QA correctness evaluations are as follows.

- **Exact Match (EM)**: Binary score of exact match between response and gold reference answer.

$$\begin{cases} 1, & \text{if generated answer exactly matches reference} \\ 0, & \text{otherwise} \end{cases}$$

- **Answer Precision**: Proportion of tokens in response that correctly appear in the gold reference answer.

$$\text{Precision} = \frac{\text{Number of overlapping tokens between generated answer and reference}}{\text{Number of tokens in generated answer}}$$

- **Answer Recall**: Proportion of tokens from the gold reference that are correctly captured in the generated answer.

$$\text{Recall} = \frac{\text{Number of overlapping tokens between generated answer and reference}}{\text{Number of tokens in reference answer}}$$

- **Token-level F1**: Harmonic mean of precision and recall.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Semantic Match**: Meaning equivalence score via pre-trained models and alignment metrics like cosine similarity between embeddings.

Popular scores that utilise or build on the above metrics are below, but are usually used as proxy metrics.

- **BLEU** [35]: calculates n-gram precision between generated answer and gold answer.
- **ROUGE** [36]: calculates n-gram recall between generated answer and gold answer.
- **MAUVE** [37]: measures distributional similarity between generated answer and gold answer.
- **FActScore** [38]: measures the proportion of generated atomic claims that are factually supported.

2.4.2 Benchmarks

Standardised benchmarks for LLM attribution are not as researched into compared to benchmarks for tasks like maths and text summarisation; however in the past few years there has been significant development.

Below are key public benchmarks used to evaluate LLM attribution and grounded generation.

AttributionBench [39] is a multi-domain benchmark, in the factoid QA task for fine-grained citations. It consists of 6 different attribution datasets which all vary in query-types, domains and difficulties to ensure comprehensiveness. For its final evaluation, it uses macro-f1 score based on a binary classification: "Attributable" and "Not Attributable". The initial research with the benchmark highlighted that LLMs lack sensitivity to fine-grained information, resulting in many of the error cases.

ALCE [22] is an open-domain QA benchmark. It focuses on evaluating fluency, correctness and citation quality and employs previously discussed metrics and scores for its evaluation. The framework investigates different retrievers and prompting techniques for any given LLM. Similarly to AttributionBench, the citation recall was limited to a binary classification of "fully supports" or "does not support" as the NLI models explored at the time for evaluation weren't good judges for "partially supports".

CAQA [26] benchmark is built upon knowledge graph question answering (KGQA). This means that the retrieval component works with structured data, rather than unstructured data that AttributionBench and ALCE work with. In a KGQA task, generated answers to queries are usually more precise and have multi-hop reasoning compared to typical text-based QA. CAQA has over 160,000 examples with varying attribution complexities to test fine-grained attribution.

Chapter 3

Experiment Setup

I have followed a similar experiment setup to experiments done in ALCE's benchmark - exploring different methods for attributed question-answering on multiple datasets.

To describe attribution formally, let q represent a query, and let $\mathcal{D}$ denote a collection of text passages. The goal is to generate an output set $\mathcal{S}$, which consists of n unique statements: $s_1, s_2, \ldots, s_n$. For each statement s_i, there is an associated set of citations $\mathcal{C}_i$, where $\mathcal{C}_i = \{c_{i,1}, c_{i,2}, \ldots\}$. Each citation $c_{i,j}$ corresponds to a passage drawn from the corpus $\mathcal{D}$.

The experimental plan is to sample multiple prediction sizes $N \in \{100, 250, 500\}$ per dataset and method to assess performance. Report correctness with dataset-appropriate metrics and evaluate citation precision and recall using a NLI model. Analyse trade-offs between metrics and test for statistically significant differences. To also explore how close we can get to prompt-integrated attribution when citations are attached post-hoc, and what the systematic gaps are.

3.1 Methods and Prompting

Three main methods are explored, in conjunction with in-context demonstrations (few-shot examples) [40]. Their details and structure are discussed in this section.

1. Post-Retrieval Answering. This method feeds a query and corresponding pre-retrieved passages to a prompt; with instructions[1] for citation and answer style (instructions are the same as ALCE or variants, depending on answer-type).

2. Post-Generation Answering (re-prompt) (w/ short or long instructions). This method requires two stages of prompting. First stage: to generate an answer to the query without any contexts. Second stage: generated answer is then inserted into a new prompt with the corresponding query and pre-retrieved passages, and instructed to add citations post-hoc. Instructions for re-prompting can be short or

[1] Exact prompts are in the github repository

long. In practical framework LangChain, this approach is called *generation post-processing* [41].

3. Post-Generation Answering (without re-prompt + TF-IDF). Only one stage of prompting needed, to generate answers from a query without contexts. The best matching passage is found for each statement via TF-IDF searching, and is then cited. GTR displays better performance than TF-IDF for this task, but TF-IDF is significantly less computationally expensive therefore is used in this research. This method provides a simpler, more deterministic post-hoc attribution compared to re-prompting.

Few-shot examples in batches of 1 or 3 are prepended to the prompt for in-context learning and are compared to zero-shot prompting. ALCE reports an increase in performance when used, so we investigate the improvements across models for the above methods.

Re-prompting can lead to unwanted changes in the final answer, even though instructions explicitly direct the model to only add citations and preserve all original content. This risk arises because the model may (i) rephrase or paraphrase parts of the answer while trying to fit in citations; (ii) reorder clauses, merge sentences, or drop nuance to reduce contradiction with retrieved passages; or (iii) infer missing context or over-correct to align with the evidence, inadvertently altering the intended meaning. Measuring the difference between initial and final answers is discussed in section 3.3, specifically how short and long instructions affect it.

Below are the prompt structures for each method.

```
{Few-shot examples} # if enabled

Instruction: {dataset-specific instruction}
Question: {question}
{pre-retrieved documents for question}
Answer:
```

Figure 3.1: Prompt structure for Post-Retrieval

```
Prompt #1 for generating answer without citations:
{Few-shot examples} # if enabled

Instruction: {dataset-specific instruction}
Question: {question}
Answer:

Prompt #2 for citation generation:
{Few-shot examples} # if enabled

Instruction: {dataset-specific instruction}
Question: {question}
Answer without citations: {answer from prompt #1}
```

```
{pre-retrieved documents for question}
Answer with citations:
```

Figure 3.2: Prompt structure for Post-Generation (w/ re-prompt)

```
{Few-shot examples} # if enabled

Instruction: {dataset-specific instruction}
Question: {question}
Answer:
```

Figure 3.3: Prompt structure for Post-Generation (w/out re-prompt + TF-IDF)

3.2 Datasets

The following datasets are used in this research.

Table 3.1: Overview of QA datasets.

Dataset	Question Origin	Corpus for retrieval	QA Type	Domain	Citation Granularity
ASQA	AmbigQA (from NQ-Open)	Wikipedia (2018-12-20)	Long-form	Open-domain	Sentence/phrase-level
ELI5	Reddit	Sphere / Web crawl	Long-form	Open-domain	Sentence/phrase-level
HAGRID	MIRACL	Wikipedia (2019-02-01)	Medium/short	Open-domain	Sentence/phrase-level
Natural Questions	Google queries	Wikipedia (2018-12-20)	Long/short	Open-domain	Sentence-level
MS MARCO	Bing queries	Web (Bing search results)	Short-form	Open-domain	Sentence-level
QAMPARI	Wikipedia	Wikipedia (2018-12-20)	List of entities	Open-domain	Entity-level

ELI5 [19] to test the ability to generate long, comprehensive, yet simple explanations for how/why/what questions. Since answers are long, multiple passages are needed as evidence. ALCE's annotated ELI5 is used as citation and correctness measures are in place. Sphere is the corpus, with each document divided into fixed-length 100-word passages. A BM25 retriever was then used to obtain top-100 passages for each query. These passages were then further processed to obtain an "oracle" set of 5 gold passages for each query, which matched recall@100 scores (how well answers contained relevant information).

ALCE also provided the oracle sets for ASQA and QAMPARI, allowing the citation task to not be affected by the performance of retrievers. The three ALCE annotated datasets extracted 1000 development examples from the originals, which is what we use.

ASQA [23] for ambiguous questions that have multiple valid long-form answers, testing the model's ability to handle nuance and disambiguation. Long-form mean answers need multiple citations. ALCE's corpus for ASQA and QAMPARI was the DPR 2018-12-20 Wikipedia snapshot (split into fixed 100-word passages) but as mentioned above, the oracle set was used.

QAMPARI [24], where answers are a long list of entities which need to be extracted from multiple passages, and hence needs many citations. Questions have at least 5 answers, averaging 13 answers per question. The ALCE version of QAMPARI was used, thus oracle set of 2018-12-20 Wikipedia was the corpus.

HAGRID [42], a dataset based on paragraphs from Wikipedia. HAGRID does not provide gold, human-annotated answers for each question, but provides GPT-3.5 generated answers with human annotations for the answers' informativeness and attributability. The corpus is a 2019-2-1 Wikipedia snapshot. It provides short and long-form answers., but only the long-form answers are used in this research.

Natural Questions (NQ) [43]. Top-5 passages from DPR [44] wikipedia 2018-12-20 was used as the corpus for citations[2]. Although it has long-form answers in the original dataset, the DPR passage retrieval version uses only examples with short-form answers, which is what is used in the experiments.

MS MARCO [45]. The web passages and documents are also included in the dataset and answers are usually a short phrase, numerical value or entity.

3.3 Evaluation Metrics

We measure correctness and citation quality for generated answers.

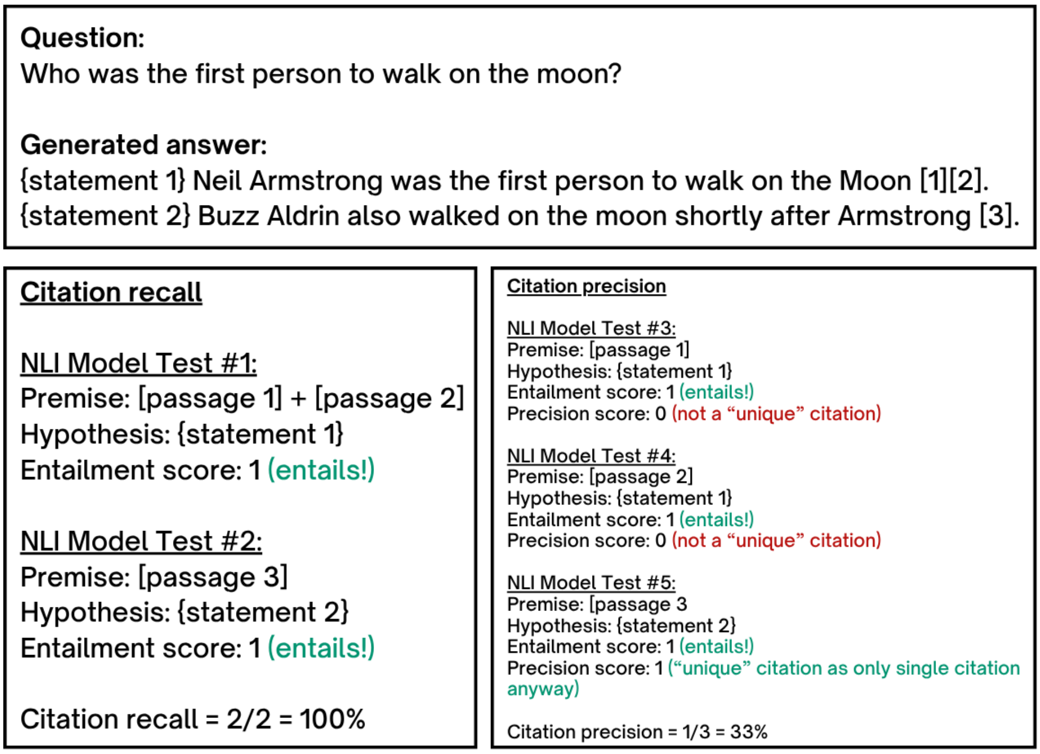
3.3.1 Citation Quality

Citation recall for each statement is defined as the proportion of statements that are appropriately supported by at least one cited source. For a given statement s_i, recall is considered to be 1 if at least one cited source is provided and the concatenation of all cited passages can fully justify the statement according to a Natural Language Inference (NLI) model. In mathematical terms, recall for s_i is 1 if both (1) the set of citations C_i is not empty and (2) the NLI model confirms that the concatenated content of those citations entails s_i. The final citation recall metric is computed by averaging over all statements in a model's output.

Citation precision assesses the relevance of citations provided for a statement. A citation is deemed irrelevant if it (1) cannot independently support the statement, and (2) its removal does not affect whether the remaining citations still fully support the statement, as determined by the NLI model. This means that a citation is only relevant if its presence is necessary to maintain the factual support for the statement. Precision is calculated for each citation and then averaged across all citations. A statement achieves perfect precision if all its citations are relevant and it satisfies recall. This approach does not penalize redundancy, recognizing that citing multiple sources can increase credibility, especially for sensitive topics such as medical claims.

To determine entailment, the same AutoAIS method as ALCE was employed, with

[2] DPR version of NQ provided by https://ir-datasets.com/



Question:
Who was the first person to walk on the moon?

Generated answer:
{statement 1} Neil Armstrong was the first person to walk on the Moon [1][2].
{statement 2} Buzz Aldrin also walked on the moon shortly after Armstrong [3].

Citation recall

NLI Model Test #1:
Premise: [passage 1] + [passage 2]
Hypothesis: {statement 1}
Entailment score: 1 (entails!)

NLI Model Test #2:
Premise: [passage 3]
Hypothesis: {statement 2}
Entailment score: 1 (entails!)

Citation recall = 2/2 = 100%

Citation precision

NLI Model Test #3:
Premise: [passage 1]
Hypothesis: {statement 1}
Entailment score: 1 (entails!)
Precision score: 0 (not a "unique" citation)

NLI Model Test #4:
Premise: [passage 2]
Hypothesis: {statement 1}
Entailment score: 1 (entails!)
Precision score: 0 (not a "unique" citation)

NLI Model Test #5:
Premise: [passage 3
Hypothesis: {statement 2}
Entailment score: 1 (entails!)
Precision score: 1 ("unique" citation as only single citation anyway)

Citation precision = 1/3 = 33%

Figure 3.4

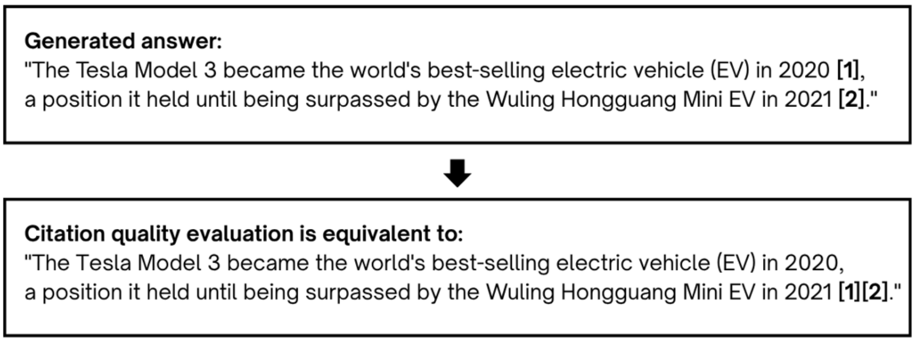
google's NLI model `t5_xxl_true_nli_mixture` [46]. The T5-based model is fine-tuned on multiple NLI datasets, to assess whether a hypothesis is entailed by a premise by computing an entailment score of either 0 (non-entailed) or 1 (entailed). In context of this task, premise is the attributed citation, and hypothesis is the claim from a generated answer. It does have a context window, so premises (cited document) need to be truncated if too long. Figure 3.4 shows how citation quality scores are computed with an example.

Although long-form answers are instructed to contain fine-grained citations at claim-level, the AutoAIS evaluation is implemented at the sentence-level for the sake of simpler implementation and alignment with ALCE's research. Figure 3.5 shows how. Citations per claim aids readability and verification for the user. A single sentence can have multiple claims, so by appending all the citations at the end of the sentence, it becomes difficult for the user to see which citation is for which claim.

Datasets differ in citation granularity. All long-form answers can have claim-level citations, but are assessed as sentence-level in the way we have implemented the NLI model.

3.3.2 Correctness

Our objective is to quantify whether a model's response is *accurate with respect to the reference answer*, using task-appropriate automatic metrics. Since answer formats vary across datasets (short entities vs. long explanations), different metrics are used



Generated answer:
"The Tesla Model 3 became the world's best-selling electric vehicle (EV) in 2020 **[1]**, a position it held until being surpassed by the Wuling Hongguang Mini EV in 2021 **[2]**."

Citation quality evaluation is equivalent to:
"The Tesla Model 3 became the world's best-selling electric vehicle (EV) in 2020, a position it held until being surpassed by the Wuling Hongguang Mini EV in 2021 **[1][2]**."

Figure 3.5

for each dataset.

For long-form answers

ASQA has ambiguous factoid questions with long-form answers, but provides a set of *short answers* representing the different interpretations for each question. The disambiguated questions and answers were provided via human annotation from AmbigQA's research [47], facilitating more reliable correctness metrics, as opposed to relying to semantic similarity scores. We follow the dataset protocol and compute short-answer recall via exact substring match against generated answers:

- **STR-EM**: fraction of gold short answers that appear as substrings in the model output;
- **STR-Hit**: 100 if all short answers are found for an instance; 0 otherwise.

ELI5 targets open-ended, multi-sentence explanations. ALCE, however, curated three simple "claims" from each reference reddit answer (using InstructGPT text-davinci-003), so we therefore evaluate *claim recall* for correctness, like ALCE. For each instance, we check whether the model output *entails* each dataset-provided claim with an NLI checker and average over claims.

HAGRID provides long, attributed answers built over MIRACL [48] passages - generated by the ChatGPT-3.5 model. There are no provided atomic claims based on the reference answers provided by HAGRID or other research, so **answer completeness**, a word-overlap calculation, is used instead. Preliminary experiments between answer completeness and other semantic similarity metrics were carried out and answer completeness was found to be most representative of correctness.

For short-form answers

For short-answer datasets, we implement dataset-specific QA metrics that directly assess factual accuracy, complemented by auxiliary semantic similarity metrics where appropriate.

Natural Questions and **MS MARCO** are factoid question-answering datasets that provide concise, specific answers to well-defined questions. Both datasets include multiple valid reference answers per question to account for lexical variations (e.g., "Paris", "paris", "Paris, France"). We implement two complementary evaluation approaches: QA and String (STR) Metrics.

QA Metrics (SQuAD-style [49] String Evaluation): Following standard factoid QA evaluation protocols, we compute exact match and F1 scores between normalized generated and reference answers:

- **QA-EM**: binary exact match after SQuAD-style normalization (lowercasing, article removal, punctuation cleaning);
- **QA-F1**: token-level overlap F1 score using precision and recall over normalized tokens;

We apply max-over-references aggregation (SQuAD convention) where the highest score across all valid reference answers is taken for each question.

STR Metrics (Substring Matching): To account for cases where models generate longer contextual answers containing the correct factual information, we compute substring-based metrics:

- **STR-EM**: binary match if any reference answer appears as a substring within the generated answer (after normalization);
- **STR-F1**: token-level overlap F1 between reference answer tokens and generated answer tokens;
- **STR-Hit**: binary indicator (100% if best F1 $>$ 0.5, 0% otherwise) measuring if the answer has sufficient token overlap.

QAMPARI targets multi-answer questions requiring lists of entities (e.g., "Name countries in Europe"). We parse comma-separated model outputs into predicted entity lists and implement set-based evaluation:

- **Recall-Top5**: modified recall that caps both hits and total answers at 5 (achieving 100% when $\geq$5 correct);
- **F1-Top5**: F1 computed using Recall-Top5 instead of standard recall.
- **Num-Preds**: number of entities predicted per question.

Model outputs are parsed as comma-separated lists after normalization. Then compute set intersection with gold answer sets to determine correctness. This approach directly measures the model's ability to provide comprehensive, accurate entity lists rather than relying on string similarity proxies.

For all short-answer datasets, the QA/STR metrics provide direct measures of factual accuracy that are more interpretable and reliable than semantic similarity scores, as they assess whether specific factual information is correctly conveyed rather than general textual overlap.

3.3.3 Preservation

Due to the non-deterministic nature of LLMs, re-prompting is expected to modify answers beyond adding citations. To measure and capture this, 3 metrics are employed.

(1) Answer-level similarity. We compute the normalized similarity between the cleaned initial and final texts using Python's `difflib.SequenceMatcher.ratio()`, returning a value in $[0, 1]$. A score of 1.0 means the two normalized strings are identical.

(2) Preservation Levenshtein (PresLev) [9].

$$\mathrm{PresLev}(x, y) \;=\; \max\left(1 - \frac{\mathrm{Lev}(x, y)}{|x|},\, 0\right),$$

Where $\mathrm{Lev}(x, y)$ is the Levenshtein edit distance between the initial text x and the stripped final text y, and $|x|$ is the length (in characters) of x. This metric lies in $[0, 1]$, with 1 indicating perfect preservation and 0 indicating full divergence or overwrite relative to the original length.

(3) Length change. We record how much the answer length changes as:

$$\Delta_{\mathrm{len}} = |y| - |x|, \quad \mathrm{ratio}_{\mathrm{len}} = \frac{|y|}{|x|}, \quad \Delta_{\%} = 100 \times \frac{|y| - |x|}{|x|}.$$

These quantify expansion or compression introduced by the citation prompt.

3.3.4 LLM Models

As half the datasets are the same as ALCE's, we try to follow similar LLMs. **gpt-3.5-turbo-0301** is primarily used for their experiments however this is now deprecated, so we use the closest model **gpt-3.5-turbo**.

Llama and Llama-instruct models were mainly used for their open-source models, so we use similar models: **Qwen2.5-14B-Instruct** and **Qwen2.5-72B-Instruct** to provide diversity in model sizes, and similar instruction-tuned behaviour.

We follow and compare our results to the corresponding GPT-3.5 baselines in their study for validity - specifically ClosedBook + PostCite (GTR) to our Post-Generation (without re-prompt + TF-IDF) and Vanilla Oracle (5-psg) to our Post-Retrieval.

Chapter 4

Experiments and Results

Due to timing and costs, only 100 examples from each dataset were tested instead of the planned subsets of 100, 250 and 500. However, results and findings that can be compared to ALCE baselines are similar for ASQA and ELI5: within 5% for citation quality and 1.5% for correctness. QAMPARI baselines, however, were significantly different to our experiments. Much higher scores than baselines were displayed, possibly due to an increase in the capability of gpt-3.5-turbo compared to the deprecated version used in ALCE. Few-shot experiments were also limited to zero and one-shot prompting.

4.1 Long-form QA trends

	Correctness	**Citation**		
Method	EM Rec.	Rec.	Prec.	F1
GPT-3.5-Turbo				
Post-Retrieval	48.6	**78.9**	**77.9**	**77.3**
Post-Generation (Short)	36.8	25.9	29.8	26.9
Post-Generation (Long)	33.7	30.3	32.5	30.3
Post-Generation (TF-IDF)	34.4	24.1	24.3	24.2
Qwen2.5-72B				
Post-Retrieval	48.0	67.6	62.3	62.5
Post-Generation (Short)	34.7	20.6	23.3	20.1
Post-Generation (Long)	35.3	20.7	23.4	20.5
Post-Generation (TF-IDF)	32.7	13.2	13.5	13.2
Qwen2.5-14B				
Post-Retrieval	**49.2**	57.4	62.1	57.6
Post-Generation (Short)	37.3	24.8	32.5	27.0
Post-Generation (Long)	29.4	19.2	26.5	20.9
Post-Generation (TF-IDF)	28.6	13.5	13.7	13.5

(a) ASQA Results

	Correctness	**Citation**		
Method	Claim Rec.	Rec.	Prec.	F1
GPT-3.5-Turbo				
Post-Retrieval	17.0	**56.3**	**55.0**	**54.8**
Post-Generation (Short)	14.8	10.2	11.3	10.2
Post-Generation (Long)	12.8	12.4	11.4	11.5
Post-Generation (TF-IDF)	14.3	9.4	9.2	9.3
Qwen2.5-72B				
Post-Retrieval	21.0	38.3	34.0	35.3
Post-Generation (Short)	22.9	6.6	8.6	6.9
Post-Generation (Long)	**24.6**	6.5	7.4	6.7
Post-Generation (TF-IDF)	22.7	5.0	5.2	5.1
Qwen2.5-14B				
Post-Retrieval	20.3	31.0	29.3	29.1
Post-Generation (Short)	21.2	6.6	6.9	6.4
Post-Generation (Long)	19.5	6.0	5.5	5.3
Post-Generation (TF-IDF)	20.0	6.3	6.2	6.2

(b) ELI5 Results

Table 4.1: ASQA and ELI5

Post-retrieval remains the strongest approach for both correctness and citation quality across long-form QA datasets. The performance gap between post-retrieval

	Correctness	**Citation**		
Method	Ans. Comp.	Rec.	Prec.	F1
GPT-3.5-Turbo				
Post-Retrieval	**84.7**	79.2	83.4	80.2
Post-Generation (Short)	72.3	38.3	44.9	39.8
Post-Generation (Long)	64.6	34.0	35.9	34.3
Post-Generation (TF-IDF)	63.8	30.9	32.1	31.3
Qwen2.5-72B				
Post-Retrieval	79.4	**83.3**	81.1	80.6
Post-Generation (Short)	66.9	21.1	22.4	20.9
Post-Generation (Long)	65.5	20.1	26.4	21.8
Post-Generation (TF-IDF)	67.8	13.4	13.9	13.5
Qwen2.5-14B				
Post-Retrieval	79.1	81.8	**85.1**	**81.7**
Post-Generation (Short)	70.1	35.7	43.7	37.3
Post-Generation (Long)	69.0	25.7	31.4	26.6
Post-Generation (TF-IDF)	62.9	17.8	18.2	17.9

Table 4.2: HAGRID Results

and post-hoc methods is consistent across all three language models tested (GPT-3.5, Qwen-72B, and Qwen-14B), demonstrating the robustness of this finding.

For post-generation, re-prompted citations consistently performed better for citation quality compared to citations via TF-IDF searching. They all have similar correctness scores as expected, since "initial" prompts for citation-less answers were under the same conditions (seed and prompt-wise). It can also be observed that

Comparing the different models, GPT-3.5 has the best citation quality metrics for ASQA and ELI5, but on HAGRID all models perform similarly - with lower parameter Qwen model having the highest citation F1. Correctness is also similar on HAGRID for all models. Qwen models, however, outperform GPT-3.5 for correctness on ASQA and ELI5. Including Qwen's lower parameter model of 7B would have helped spotting a clearer trend with HAGRID.

4.2 Short-form QA trends

Post-retrieval remains strongest across all short-form datasets (QAMPARI, MS MARCO, Natural Questions). It delivers the best citation quality and correctness - often by a wide margin.

Long instructions for post generation re-prompting consistently improves citation quality on QAMPARI and MS MARCO, but not for Natural Questions. Natural Questions has the shortest answers of all the datasets, hence may find longer instructions for citation noisy and excessive. Post generation with TF-IDF searching outperforms re-prompting on MS MARCO for citation quality.

QAMPARI. For post-generation, long-instructions improve citation F1 vs short for every model (GPT-3.5: 30.1 → 30.3; Qwen-72B: 16.6 → 23.1; Qwen-14B: 19.2 → 20.4). Gains are largest on Qwen-72B. TF-IDF trails in comparison to re-prompting on citation quality. Changes in correctness differ per model, GPT-3.5 and Qwen-72B preferring long instructions over short (for recall top-5) and Qwen-14B preferring short.

	Correctness					**Citation**		
Method	QA-EM	QA-F1	QA-Hit	STR-EM	STR-F1	Rec.	Prec.	F1
GPT-3.5-Turbo								
Post-Retrieval	**72.0**	**79.0**	**80.0**	78.0	79.0	**48.0**	**48.0**	**48.0**
Post-Generation (Short)	46.0	55.3	50.0	58.0	55.4	33.5	34.0	33.7
Post-Generation (Long)	44.0	51.3	48.0	52.0	51.6	29.2	29.3	29.2
Post-Generation (TF-IDF)	46.0	55.5	52.0	54.0	55.8	25.0	25.0	25.0
Qwen2.5-72B								
Post-Retrieval	62.0	69.3	68.0	74.0	69.5	44.0	44.0	44.0
Post-Generation (Short)	30.0	39.5	34.0	72.0	40.2	24.8	27.5	25.5
Post-Generation (Long)	32.0	39.8	36.0	62.0	40.3	18.7	20.0	19.0
Post-Generation (TF-IDF)	30.0	36.8	34.0	44.0	36.9	14.0	14.0	14.0
Qwen2.5-14B								
Post-Retrieval	68.0	75.1	74.0	**80.0**	**75.3**	44.0	44.0	44.0
Post-Generation (Short)	18.0	30.5	20.0	76.0	31.5	28.4	33.2	29.9
Post-Generation (Long)	34.0	45.1	42.0	64.0	45.5	27.0	32.3	28.3
Post-Generation (TF-IDF)	14.0	21.8	18.0	30.0	22.0	11.7	11.8	11.7

Table 4.3: Natural Questions Results

	Correctness					**Citation**		
Method	QA-EM	QA-F1	QA-Hit	STR-EM	STR-F1	Rec.	Prec.	F1
GPT-3.5-Turbo								
Post-Retrieval	14.0	**44.6**	**44.0**	28.0	**46.2**	13.5	13.0	12.9
Post-Generation (Short)	0.0	23.3	8.0	32.0	25.6	14.3	14.4	14.2
Post-Generation (Long)	2.0	20.4	12.0	24.0	21.5	17.7	11.6	13.5
Post-Generation (TF-IDF)	6.0	19.3	12.0	8.0	20.1	20.2	20.2	20.2
Qwen2.5-72B								
Post-Retrieval	**20.0**	38.4	30.0	28.0	39.0	**22.2**	**17.8**	**19.3**
Post-Generation (Short)	2.0	23.0	8.0	20.0	24.3	13.9	9.7	10.9
Post-Generation (Long)	8.0	23.3	12.0	12.0	23.9	15.7	11.5	12.9
Post-Generation (TF-IDF)	6.0	20.4	8.0	10.0	20.9	18.0	18.1	18.0
Qwen2.5-14B								
Post-Retrieval	**20.0**	44.8	38.0	**36.0**	45.8	20.0	15.8	17.2
Post-Generation (Short)	0.0	18.4	2.0	28.0	19.7	9.2	8.3	8.4
Post-Generation (Long)	8.0	25.9	16.0	18.0	26.7	14.5	9.8	11.3
Post-Generation (TF-IDF)	6.0	18.7	10.0	6.0	19.3	13.5	13.5	13.5

Table 4.4: MS MARCO Results

	Correctness		Citation		
Method	Rec.-5	Prec.	Rec.	Prec.	F1
GPT-3.5-Turbo					
Post-Retrieval	**44.0**	**43.9**	**73.8**	**75.2**	**74.0**
Post-Generation (Short)	13.5	18.3	30.1	30.6	30.1
Post-Generation (Long)	15.8	24.5	30.1	30.9	30.3
Post-Generation (TF-IDF)	13.4	18.1	23.0	24.4	23.4
Qwen2.5-72B					
Post-Retrieval	42.0	46.1	69.8	71.5	70.2
Post-Generation (Short)	22.6	16.9	15.7	19.3	16.6
Post-Generation (Long)	21.6	18.5	22.0	26.4	23.1
Post-Generation (TF-IDF)	21.6	18.1	18.8	18.8	18.8
Qwen2.5-14B					
Post-Retrieval	40.2	43.7	63.7	70.6	65.3
Post-Generation (Short)	12.7	10.6	17.1	26.2	19.2
Post-Generation (Long)	13.7	14.3	17.8	29.4	20.4
Post-Generation (TF-IDF)	15.2	13.1	11.1	11.1	11.1

Table 4.5: QAMPARI Results

MS MARCO. TF-IDF is the standout for citation quality across post-generations. For re-prompting, changing from short to long instructions in post-generation helps Qwen models a bit (72B: 10.9 → 12.9; 14B: 8.4 → 11.3) but hurts GPT-3.5 (14.2 → 13.5). Correctness is mixed and smaller than the retrieval gap.

Natural Questions. For post-generation re-prompting, short instructions beat long on citation F1 for all models (GPT-3.5: 33.7 → 29.2; Qwen-72B: 25.5 → 19.0; Qwen-14B: 29.9 → 28.3).

4.3 Short vs. long instructions in re-prompt

The effect of instruction length in re-prompting methods show model-dependent and dataset-dependent patterns. In the case of GPT-3.5, longer instructions improved citation quality for ASQA and ELI5, but worsened for HAGRID. Qwen's 72B had almost identical results between short and long instructions (largest F1 difference of 0.9% in HAGRID), whereas the 14B model interestingly had better results on all long-form datasets with short instructions. An explanation for this could be that longer instructions are noisy for smaller models, so they prioritise certain steps of the instruction over others [50] [51]. To explore this further, a comparison between the initial generated answers and final cited answers is done.

Table 4.6: Comparison between initial answers and final cited answers (post citation-stripping) for short and long instructions. Identical % = percentage of identical initial and final answers. PresLev = Levenshtein Preservation. % Δ Length = % length change.

Model	Dataset (Mode)	Identical %	PresLev	Δ Length %
GPT-3.5-Turbo				
	ASQA (Short)	75.8	0.909	7.5
	ASQA (Long)	57.6	0.854	-8.2
	ELI5 (Short)	79.8	0.980	-0.3
	ELI5 (Long)	60.6	0.886	-4.1
	HAGRID (Short)	33.0	0.578	122.0
	HAGRID (Long)	52.0	0.862	14.9
	MSMARCO (Short)	3.0	0.037	2430.7
	MSMARCO (Long)	13.1	0.175	818.9
	Natural Questions (Short)	79.0	0.864	56.9
	Natural Questions (Long)	77.0	0.840	4.7
	QAMPARI (Short)	64.6	0.817	48.1
	QAMPARI (Long)	78.8	0.939	-4.3
Qwen-72B				
	ASQA (Short)	32.3	0.841	4.8
	ASQA (Long)	44.4	0.893	-6.7
	ELI5 (Short)	35.4	0.884	0.9
	ELI5 (Long)	43.4	0.911	-8.9
	HAGRID (Short)	23.0	0.729	10.9
	HAGRID (Long)	34.0	0.808	0.9
	MSMARCO (Short)	51.5	0.636	482.4
	MSMARCO (Long)	87.9	0.934	13.9
	Natural Questions (Short)	59.0	0.609	733.3
	Natural Questions (Long)	73.0	0.730	611.9
	QAMPARI (Short)	36.4	0.509	160.9
	QAMPARI (Long)	57.6	0.783	33.7
Qwen-14B				
	ASQA (Short)	46.5	0.713	23.9
	ASQA (Long)	67.7	0.900	3.2
	ELI5 (Short)	53.5	0.886	0.5
	ELI5 (Long)	55.6	0.913	-5.4
	HAGRID (Short)	31.0	0.590	49.5
	HAGRID (Long)	48.0	0.759	25.5
	MSMARCO (Short)	0.0	0.060	2813.8
	MSMARCO (Long)	23.2	0.439	467.3
	Natural Questions (Short)	16.0	0.203	1125.3
	Natural Questions (Long)	45.0	0.512	737.0
	QAMPARI (Short)	10.1	0.339	192.0
	QAMPARI (Long)	25.3	0.609	146.0

Table 4.6 shows that for both methods, the re-prompting changes the original answer in a significant amount of cases, despite being instructed not to. Longer citation instructions tend to preserve the original answers more and reducing aggressive rewrites. For GPT-3.5, long instructions cut average length inflation substantially on HAGRID ($122\% \to 15\%$), MS MARCO ($2431\% \to 819\%$), Natural Questions ($57\% \to$

4.7%), and QAMPARI ($48\% \rightarrow -4.3\%$), while raising or matching PresLev in the same cases (e.g., HAGRID $0.58 \rightarrow 0.86$). Qwen-72B shows the same direction of effect across *all* datasets (long $>$ short in PresLev), indicating that the longer prompt stabilizes editing rather than prompting a fresh re-write.

Qwen-14B despite sometimes achieving better citation F1 with short instructions on long-form QA, preserved text better with long and avoids the extreme spikes in added information seen with short instructions on MS MARCO/ Natural Questions. This suggests citation quality improved because re-prompt generated answers wanted to align more with the pre-retrieved documents. This is further supported by the fact that correctness improved with short instructions compared to long. Figure 4.1 shows where most identical generations occurred. GPT-3.5 had the most identical answers, then Qwen-72B, and lastly Qwen-14B. All long-form datasets had the same correlation across models, but the short-form datasets displayed a less obvious trend. For example, on MS MARCO, GPT-3.5 keep only 8.1% of answers, whereas Qwen-72B managed to keep 69.7%. Figure 4.2 displays the distribution of PresLev scores for each model when averaged across both approaches.

4.4 One-shot prompting

One-shot prompting. Citation quality and correctness benefited from one-shot examples, especially on short-form QA datasets (Figure 4.3). When compared to zero-shot prompting, across short-form datasets, gains were significantly large (citation recall 91.2%, correctness +234.2%). For long-form datasets, gains were more modest (citation recall +18.9%, correctness +15.0%). This pattern is consistent with short answers being more focused and easier to imitate from examples, yielding clearer citation placement. Post-retrieval shows the most reliable improvements and post-generation effects depend on the re-prompting type.

One-shot prompting improved the percentage of identical answers and PresLev score between initial and final answers for both long-instructions and short. For short instructions, however one-shot examples had a very small average increase of all under 5%. Long instructions, on the other hand, had significant improvements, shown in Figure 4.4.

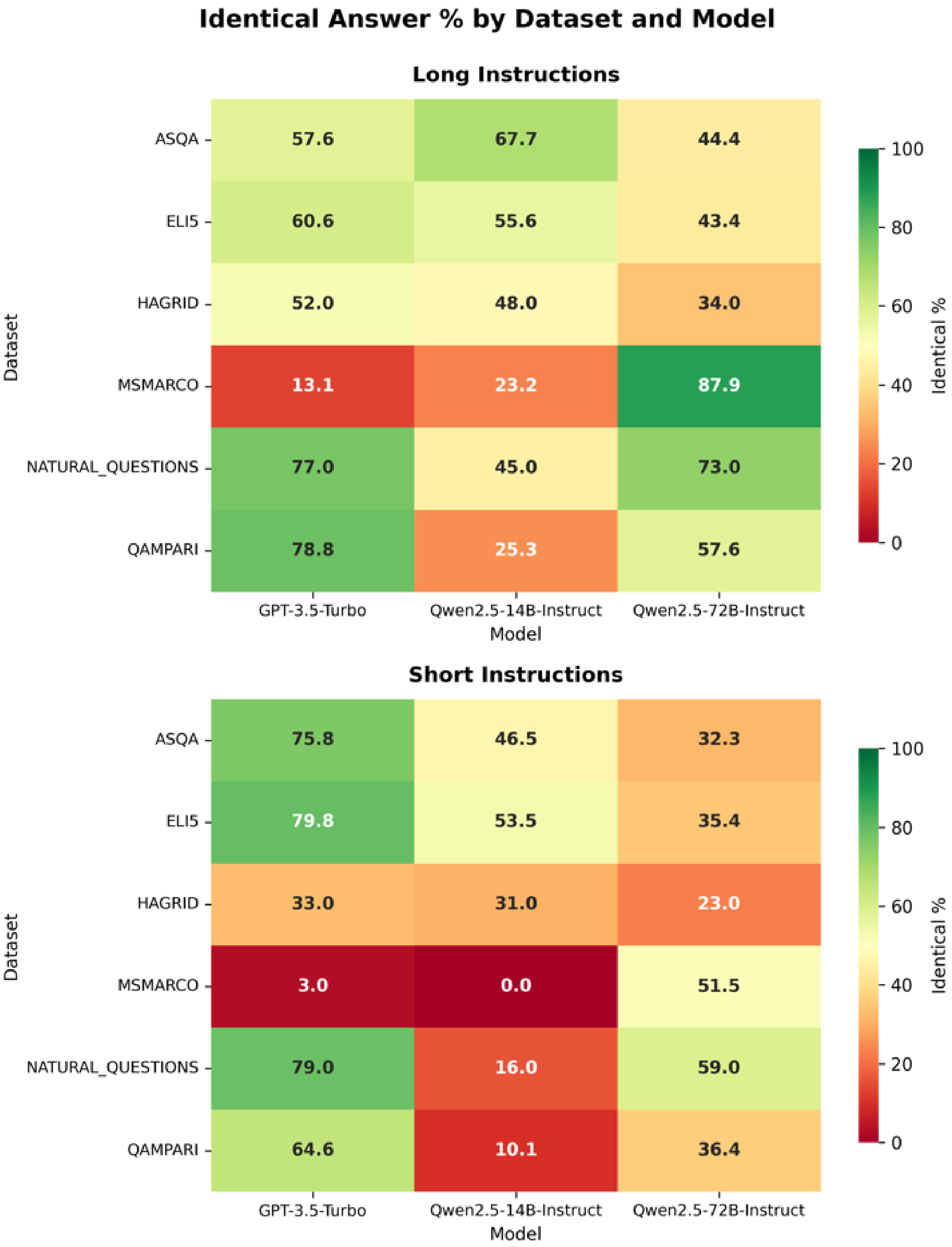

Identical Answer % by Dataset and Model
Long Instructions
Dataset
ASQA
ELI5
HAGRID
MSMARCO
NATURAL_QUESTIONS
QAMPARI
57.6
67.7
44.4
60.6
55.6
43.4
52.0
48.0
34.0
13.1
23.2
87.9
77.0
45.0
73.0
78.8
25.3
57.6
GPT-3.5-Turbo
Qwen2.5-14B-Instruct
Qwen2.5-72B-Instruct
Model
Identical %
100
80
60
40
20
0
Short Instructions
Dataset
ASQA
ELI5
HAGRID
MSMARCO
NATURAL_QUESTIONS
QAMPARI
75.8
46.5
32.3
79.8
53.5
35.4
33.0
31.0
23.0
3.0
0.0
51.5
79.0
16.0
59.0
64.6
10.1
36.4
GPT-3.5-Turbo
Qwen2.5-14B-Instruct
Qwen2.5-72B-Instruct
Model
Identical %
100
80
60
40
20
0

Figure 4.1

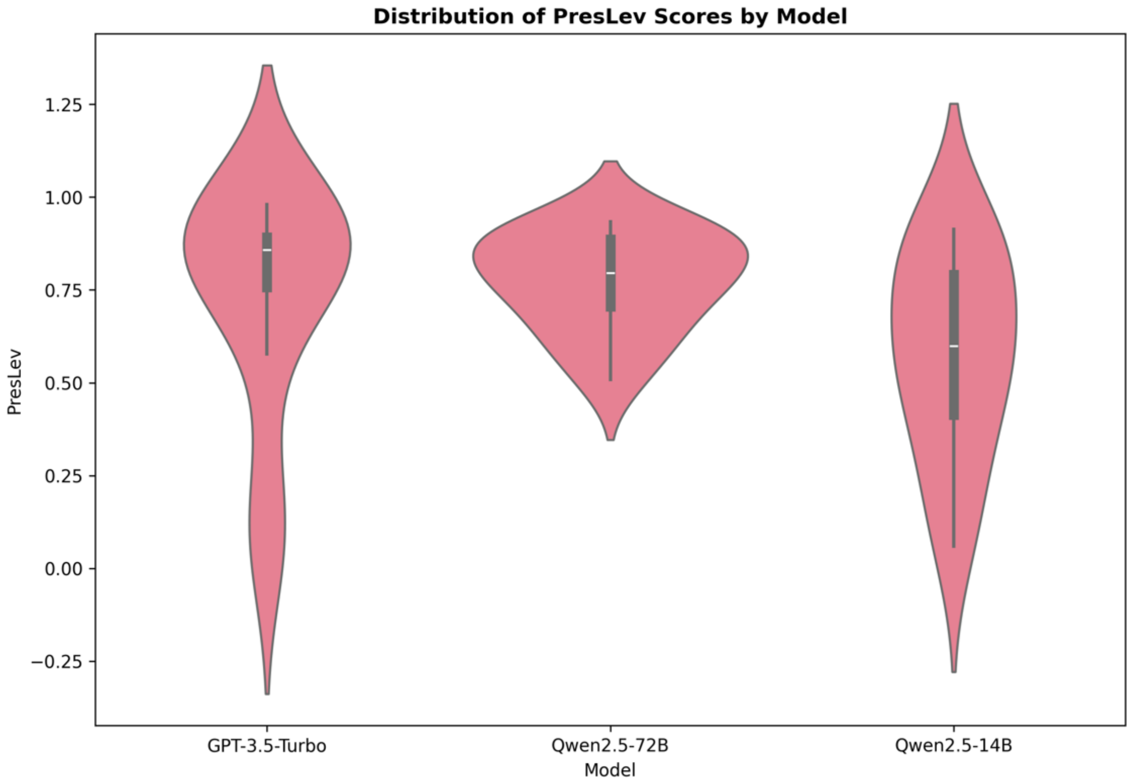


Figure 4.2

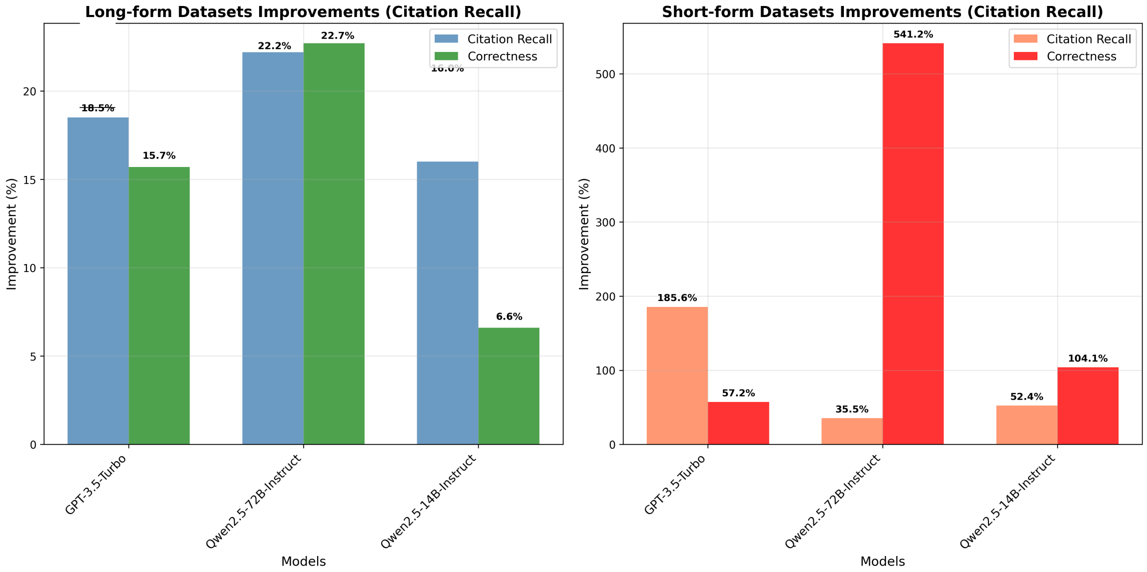


Figure 4.3: Percentage increase in citation recall and correctness when using one-shot examples

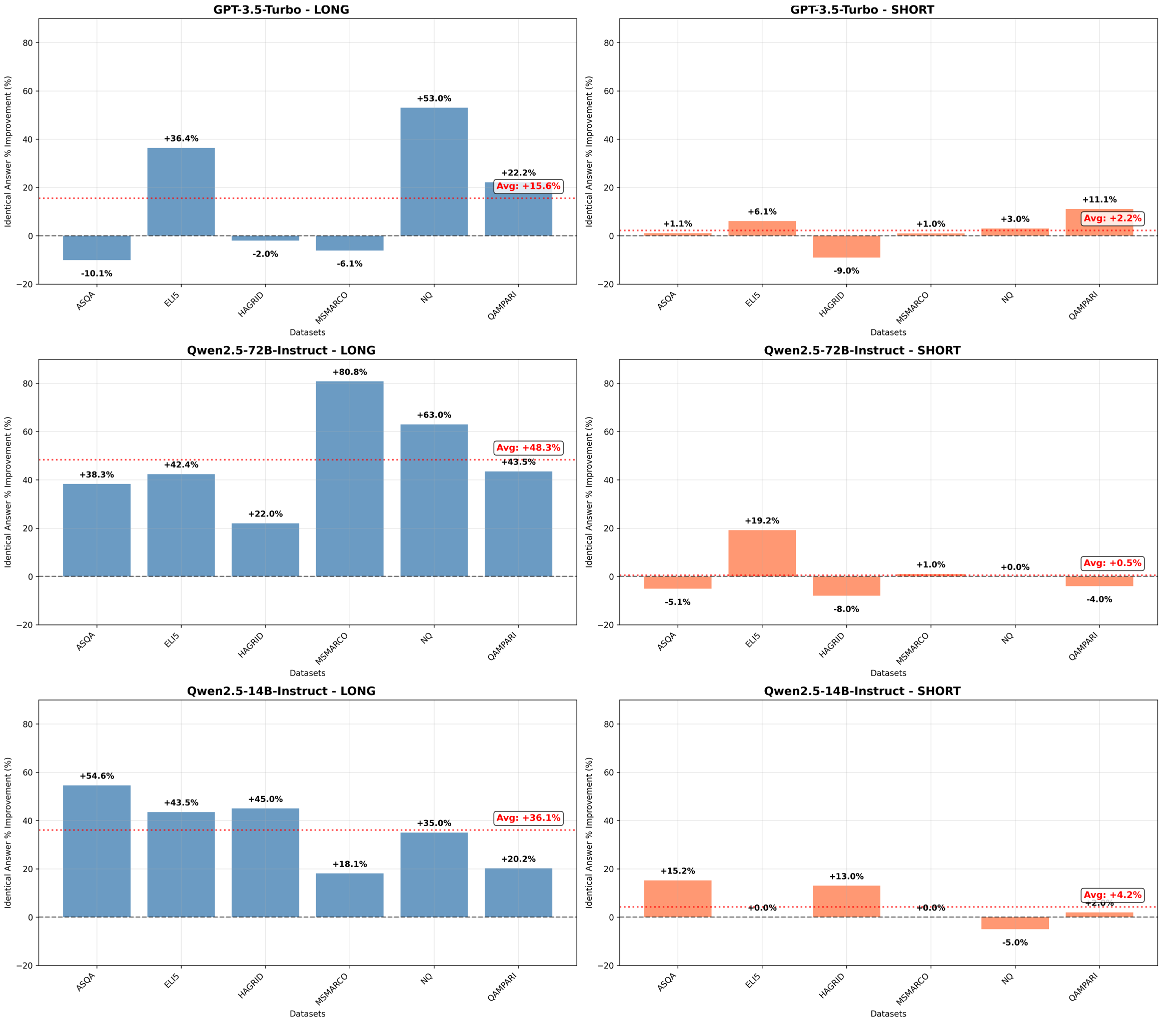


Figure 4.4: Percentage increase in identical final and initial answers per dataset for long and short instructions when using one-shot examples

Chapter 5

Conclusion

5.1 Findings

The aim of the study was to explore several attribution methods and report on their evaluations and trade-offs.

We find that Post-Retrieval is a simple, effective and robust method for attributed QA. It translates well with both short and long-form QA, and to small, open-source models.

Post-Generation methods for attribution which involve re-prompting are noisy and not reliable if solely used for citation instructions only. Despite improving on metrics compared to post-hoc TF-IDF citation, we find the final answers generated to be not aligned with the instructions given in many cases (e.g. Qwen2.5-14B-Instruct changing the final answer from initial 100% of the time on MS MARCO). Although it may be useful for users needing verifiable information, it is not reproducible and should be avoided if incorporating into a more sophisticated strategy for post-hoc attribution.

Post-Generation attribution via TF-IDF searching demonstrates to be weak consistently, even for basic short-form QA datasets like Natural Questions which only require a singe citation per answer.

In-context demonstrations significantly improves correctness and citation quality metrics for short-form QA datasets. Long-form also shows improvement, but not as dramatically.

5.2 Limitations and Future Work

The study has several limitations that need to be considered for future work.

Finer-grained citation evaluation would have highlighted more aspects of long-form QA attribution than the sentence-level AutoAIS evaluation employed in this study. ALiiCE [52] would have been an easy-to-implement possibility.

Binary classification of citation entailment is another limiting factor of the research, as our NLI model does not detect partial support from evidences to claims. This reduces nuanced results that could have been observed, as smaller models may have shown better evaluations if considered.

Using more examples would have greatly increased the reliability of the findings.

Generating sub-claims for HAGRID answers could have been explored, similar to ELI5, for a better possible correctness metric. ALCE detailed the steps taken to derive the claim recall approach they had for ELI5 correctness - using `text-davinci-003` to generate 3 sub-claims each. HAGRID shared similarities to ELI5, so could have been investigated to improve validity.

Bibliography

[1] Anqi Shao. Beyond misinformation: A conceptual framework for studying ai hallucinations in (science) communication. *arXiv preprint arXiv:2504.13777*, April 2025.

[2] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[3] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022.

[4] Nourhan Ibrahim, Samar AboulEla, Ahmed Fakhri Ibrahim, and Rasha Kashef. A survey of knowledge enhanced pre-trained language models. *Electronics*, 14(11):2102, 2024.

[5] Klaudia Jaźwińska and Aisvarya Chandrasekar. Ai search has a citation problem: We compared eight ai search engines. they're all bad at citing news. Columbia Journalism Review, March 6 2025.

[6] Nelson F. Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. *Findings of EMNLP 2023*, Findings of EMNLP, 2023. Preprint available at arXiv:2304.09848.

[7] Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. A survey of large language models attribution, 2023.

[8] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models, 2023.

[9] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. Rarr: Researching and revising what language models say, using language models, 2023.

[10] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022.

[11] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, 2022.

[12] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes, 2022.

[13] Cheng Wang, Xinyang Lu, See-Kiong Ng, and Bryan Kian Hsiang Low. Trace: Transformer-based attribution using contrastive embeddings in llms, 2024.

[14] Zhiyu Zhu, Jiayu Zhang, Zhibo Jin, Fang Chen, and Jianlong Zhou. Abe: A unified framework for robust and faithful attribution-based explainability, 2025.

[15] Renjun Xu and Jingwen Peng. A comprehensive survey of deep research: Systems, methodologies, and applications, 2025.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[17] Guido Zuccon, Bevan Koopman, and Razia Shaik. Chatgpt hallucinates when attributing answers, 2023.

[18] Junwei Deng, Ting-Wei Li, Shichang Zhang, and Jiaqi Ma. Efficient ensembles improve training data attribution, 2024.

[19] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering, 2019.

[20] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022.

[21] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017.

[22] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations, 2023.

[23] Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. Asqa: Factoid questions meet long-form answers, 2023.

[24] Samuel Joseph Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. Qampari: An open-domain question answering benchmark for questions with many answers from multiple paragraphs, 2023.

[25] Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. Automatic evaluation of attribution by large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4615–4635, Singapore, December 2023. Association for Computational Linguistics.

[26] Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Hongru Wang, Sheng Bi, Yongrui Chen, Tongtong Wu, and Jeff Z. Pan. Can llms evaluate complex attribution in qa? automatic benchmarking using knowledge graphs, 2025.

[27] Denis Peskoff and Brandon Stewart. Credible without credit: Domain experts assess generative language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 427–438, Toronto, Canada, July 2023. Association for Computational Linguistics.

[28] Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. "according to ...": Prompting language models improves quoting from pre-training data, 2024.

[29] Dongyub Lee, Taesun Whang, Chanhee Lee, and Heuiseok Lim. Towards reliable and fluent large language models: Incorporating feedback learning loops in qa systems, 2023.

[30] Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. Llatrieval: Llm-verified retrieval for verifiable generation, 2024.

[31] Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks, 2024.

[32] Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. Complex claim verification with evidence retrieved in the wild, 2024.

[33] Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. Longcite: Enabling llms to generate fine-grained citations in long-context qa, 2024.

[34] Yumo Xu, Peng Qi, Jifan Chen, Kunlun Liu, Rujun Han, Lan Liu, Bonan Min, Vittorio Castelli, Arshit Gupta, and Zhiguo Wang. Citeeval: Principle-driven citation evaluation for source attribution, 2025.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[36] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[37] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers, 2021.

[38] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation, 2023.

[39] Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. Attributionbench: How hard is automatic attribution evaluation?, 2024.

[40] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[41] LangChain. How to get a rag application to add citations: generation post-processing. https://python.langchain.com/docs/how_to/qa_citations/#generation-post-processing.

[42] Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution, 2023.

[43] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[44] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering, 2020.

[45] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.

[46] Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States, July 2022. Association for Computational Linguistics.

[47] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online, November 2020. Association for Computational Linguistics.

[48] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Making a miracl: Multilingual information retrieval across a continuum of languages, 2022.

[49] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[50] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[51] Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models, 2024.

[52] Yilong Xu, Jinhua Gao, Xiaoming Yu, Baolong Bi, Huawei Shen, and Xueqi Cheng. Aliice: Evaluating positional fine-grained citation generation, 2024.

Declarations

Use of Generative AI

I acknowledge the use of ChatGPT-5 (OpenAI, https://chatgpt.com) to generate outlines for background study and code support. I confirm that no content generated by AI has been presented as my own work.

Ethical Considerations

This research did not involve human participants, personal data, or sensitive material, and therefore did not require formal ethical approval.

Sustainability

The research was carried out with consideration for environmental sustainability. Experiments were run on shared, cloud resources (Hugging Face Spaces) with dataset sampling to minimise computational load. All unnecessary large-scale model runs were avoided to reduce carbon footprint.

Availability of Data and Materials

Source code developed and datasets used for the project has been deposited in a public google drive folder at:

https://drive.google.com/file/d/1GSyI_Q2uLcJm621hzaUKw_X2P6Z9A8Pg/view?usp=sharing