LAQuer: Localized Attribution Queries in Content-grounded Generation

Eran Hirsch[1] **Aviv Slobodkin**[1] **David Wan**[2]
Elias Stengel-Eskin[2] **Mohit Bansal**[2] **Ido Dagan**[1]

[1]Bar-Ilan University [2]UNC Chapel Hill

{hirsch.eran, lovodkin93}@gmail.com
{davidwan, esteng, mbansal}@cs.unc.edu dagan@cs.biu.ac.il

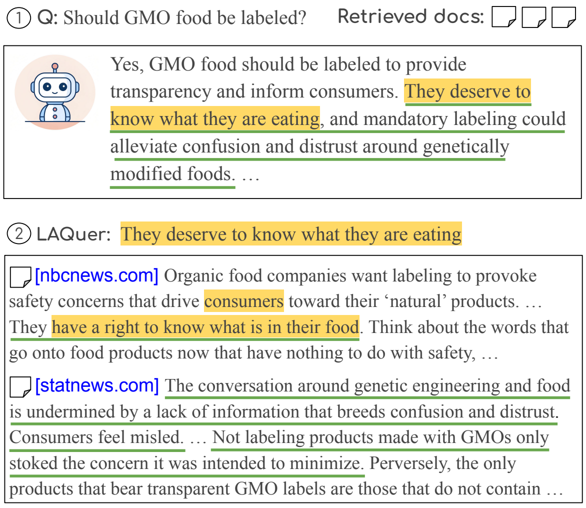
Abstract

Grounded text generation models often produce content that deviates from their source material, requiring user verification to ensure accuracy. Existing attribution methods associate entire sentences with source documents, which can be overwhelming for users seeking to fact-check specific claims. In contrast, existing sub-sentence attribution methods may be more precise but fail to align with users' interests. In light of these limitations, we introduce **L**ocalized **A**ttribution **Quer**ies (LAQuer), a new task that localizes selected spans of generated output to their corresponding source spans, allowing fine-grained and user-directed attribution. We compare two approaches for the LAQuer task, including prompting large language models (LLMs) and leveraging LLM internal representations. We then explore a modeling framework that extends existing attributed text generation methods to LAQuer. We evaluate this framework across two grounded text generation tasks: Multi-document Summarization (MDS) and Long-form Question Answering (LFQA). Our findings show that LAQuer methods significantly reduce the length of the attributed text. Our contributions include: (1) proposing the LAQuer task to enhance attribution usability, (2) suggesting a modeling framework and benchmarking multiple baselines, and (3) proposing a new evaluation setting to promote future research on localized attribution in content-grounded generation.[1]

"ChatGPT can make mistakes. Check important information." — ChatGPT interface

1 Introduction

Grounded text generation aims to produce content based on specific sources, whether retrieved—such as in retrieval-augmented generation (RAG) (Lewis et al., 2020; Ram et al., 2023)—or user-provided. Yet, model outputs frequently diverge from these sources, resulting in factual inaccuracies, or 'hallucinations' (Mishra et al., 2024). To address this, users often need to manually review retrieved documents to ensure the accuracy of generated claims. This in turn has driven a growing interest in *attributed* text generation (Thoppilan et al., 2022; Menick et al., 2022; Bohnet et al., 2023), which incorporates supporting evidence or citations into the output, thereby enhancing model reliability and helping mitigate potential factuality errors.


① Q: Should GMO food be labeled? Retrieved docs:

Yes, GMO food should be labeled to provide transparency and inform consumers. They deserve to know what they are eating, and mandatory labeling could alleviate confusion and distrust around genetically modified foods. …

② LAQuer: They deserve to know what they are eating

[nbcnews.com] Organic food companies want labeling to provoke safety concerns that drive consumers toward their 'natural' products. … They have a right to know what is in their food. Think about the words that go onto food products now that have nothing to do with safety, …

[statnews.com] The conversation around genetic engineering and food is undermined by a lack of information that breeds confusion and distrust. Consumers feel misled. … Not labeling products made with GMOs only stoked the concern it was intended to minimize. Perversely, the only products that bear transparent GMO labels are those that do not contain …

Figure 1: **Top**: example RAG scenario. **Bottom**: our Localized Attribution Queries (LAQuer), where the attribution is constructed per user query, highlighted in yellow. Existing sentence-level attribution methods, underlined in green, can often be disorienting and lengthy.

While attributed text generation enhances transparency by providing citations, its effectiveness depends on how easily users can interpret these attributions, as shown in Fig. 1. Most existing attribution methods associate each generated *sentence* with its corresponding attributions (Gao et al., 2023b; Slobodkin et al., 2024). For example, the output sentence underlined green is attributed to many spans in the source document, also under-

[1] https://github.com/eranhirs/LAQuer

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15355–15370
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

lined green. Yet, in practice, users often seek to fact-check specific details rather than an entire sentence (e.g., the highlighted fact in Fig. 1). As sentences typically contain multiple facts (Min et al., 2023), sentence-level attribution requires readers to examine both the full sentence and its sources before assessing factual accuracy of a single fact. For instance, in Fig. 1, the highlighted fact is attributed by the first source, while another within the same sentence is linked to the second source. As a result, users must review the entire sentence and all cited sources to verify this single fact.

In this work, we introduce a more precise attributed generation task, which we call **L**ocalized **A**ttribution **Quer**ies (LAQuer), that links specific spans in generated text to their corresponding source spans. Each query consists of pre-selected output spans, or 'highlights' (e.g., the highlighted span in the top part of Fig. 1), while the response identifies the relevant source spans (e.g., the highlighted spans in the bottom part of Fig. 1). Since queries can vary from single words to full sentences, this approach generalizes existing attribution methods while enabling targeted attribution.

We model the LAQuer setting as a framework consisting of two processing stages, illustrated in Fig. 2. First, a source-grounded generation system produces text expected to be supported by identified source texts. Some generation methods may include attribution metadata, mapping output segments to supporting source spans. For example, in Fig. 1, a sentence-level attribution method can attribute the second sentence to the texts underlined green. In our experiments (Section 5), we benchmark LAQuer using three generation approaches: one without attribution and two contemporary attributed-generation methods. In the second stage, users request localized attribution by highlighting spans that correspond to a fact of interest. The LAQuer task then identifies the exact supporting source spans for the given highlight. This second stage is composed of two steps: (A) decontextualization of the user's query, and (B) query-focused attribution. The decontextualization step converts the highlighted fact to a stand-alone decontextualized statement, for which source attribution can be more easily sought in an unambiguous matter. For example, "*They*" in Fig. 1 refers to "*consumers*". In such cases, attributions should account for the decontextualized meaning, e.g., that "*They*" is correctly attributed to "*consumers*." The query-focused attribution step searches for the supporting source spans for the decontextualized statement. For the query-focused attribution, we compare two approaches: one that prompts a large language model (LLM) to produce the alignment and another that uses the internal representations of the model to align phrases (Phukan et al., 2024). If attribution metadata from the generation step is available, it is leveraged to narrow the search space. For example, instead of scanning the entire source document in the figure, our approach can focus on the spans underlined green.

For evaluation, to simulate user interaction in this process, our methodology involves decomposing the generated output into atomic facts using LLMs (Min et al., 2023), which are subsequently aligned with output spans. The LAQuer task can then be applied to any type of generation, unlike previous work which focuses on datasets annotated with sub-sentence alignments (Phukan et al., 2024; Qi et al., 2024; Cohen-Wang et al., 2024). Our experimental setup includes two grounded generation tasks, Multi-document Summarization (MDS) and Long-form Question Answering (LFQA). A key finding is that LAQuer methods can significantly reduce the length of the attributed text. Overall, LAQuer remains a challenging task, particularly in attributing decontextualized facts. In total, our contribution in this work is enumerated as follows:

1. We propose Localized Attribution Queries (LAQuer) as a task to improve the accessibility of attributions for users.

2. We introduce a novel modeling framework for the LAQuer setting and benchmark various baselines. We demonstrate their potential to enable targeted attribution while maintaining accuracy.

3. We establish a new evaluation setting that encourages future research on localized attribution in content-grounded generation.

2 Background

Hallucinations produced by LLMs have attracted increasing interest in generating attributed text. The task of *attributed text generation* requires models to generate summaries or answers that cite specific evidence for their claims (Gao et al., 2023b; Thoppilan et al., 2022; Menick et al., 2022; Bohnet et al., 2023). When considering the granularity of

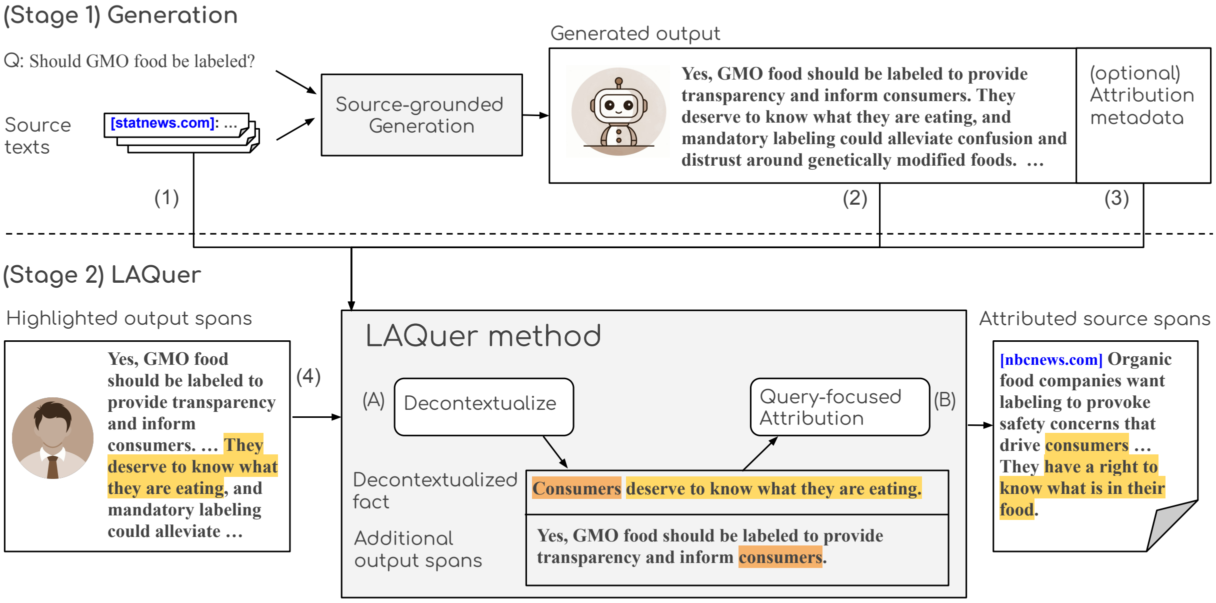

(Stage 1) Generation
Q: Should GMO food be labeled?
Source texts
[statnews.com]: ...
Source-grounded Generation
Generated output
Yes, GMO food should be labeled to provide transparency and inform consumers. They deserve to know what they are eating, and mandatory labeling could alleviate confusion and distrust around genetically modified foods. ...
(optional) Attribution metadata
(1)
(2)
(3)
(Stage 2) LAQuer
Highlighted output spans
Yes, GMO food should be labeled to provide transparency and inform consumers. ... They deserve to know what they are eating, and mandatory labeling could alleviate ...
(4)
LAQuer method
(A)
Decontextualize
Query-focused Attribution
(B)
Decontextualized fact
Consumers deserve to know what they are eating.
Additional output spans
Yes, GMO food should be labeled to provide transparency and inform consumers.
Attributed source spans
[nbcnews.com] Organic food companies want labeling to provoke safety concerns that drive consumers ... They have a right to know what is in their food.

Figure 2: Overview of our LAQuer framework. The top section illustrates the generation of an output based on identified source texts, either provided as input or retrieved. The bottom section represents the LAQuer task, where output spans are attributed back to their source texts, enabling users to verify the provenance of individual pieces of information. The inputs to our proposed LAQuer approach are labeled (1) to (4). In Step (A), the highlighted spans are transformed into a decontextualized fact along with its corresponding output spans. In Step (B), the user's query is attributed to relevant source texts, enabling precise fact verification.

the attribution, there are two key factors: the granularity of the summary or answer (i.e., the output) and the granularity of the source text (i.e., the input). The standard level of output granularity is sentence-level (Gao et al., 2023b; Slobodkin et al., 2024). Some work focuses on sub-sentence attribution, based on the internal representations of a model (Phukan et al., 2024; Qi et al., 2024; Ding et al., 2024) or manipulation to the input (Cohen-Wang et al., 2024). Similarly, input granularity can vary between pointing to the entire response (Thoppilan et al., 2022), documents (Gao et al., 2023b), snippets (Menick et al., 2022), paragraphs or sentences (Buchmann et al., 2024), and spans (Schuster et al., 2024; Phukan et al., 2024; Qi et al., 2024; Ding et al., 2024; Cohen-Wang et al., 2024).

The above methods provide fixed predetermined attributions, that often do not correspond most effectively to the specific scope of output information for which attribution is sought. Some systems provide attributions for longer output spans, requiring the user to examine irrelevant source segments (Gao et al., 2023b; Slobodkin et al., 2024), while others provide only partial attributions for narrow output spans, requiring the user to look around the attributed source spans for complete supporting information (Phukan et al., 2024; Qi et al., 2024; Ding et al., 2024). Our work is the first to explore user-initiated attribution queries across variable scales, introducing a novel evaluation methodology to assess their effectiveness.

3 Localized Attribution Queries

The LAQuer task assumes as input a generation o grounded in source documents D. For instance, in Fig. 1, the answer to the question *"Should GMO food be labeled?"* is generated based on two source documents. A key aspect of this task is the inclusion of 'highlights', which are specific parts of the generated output that are marked by the user. These highlights indicate a fact that the user wants to verify or examine within the source. The user conveys the fact of interest by selecting the spans in the output that best express it. For example, in the figure the highlighted span is: *"They deserve to know what they are eating."* Importantly, the user may not care about other claims made in the same sentence, such as *"labeling could alleviate confusion and distrust."* Formally, we are given a set of highlighted output spans $o_1, \ldots, o_n$ where each span may range in length from a single word to the entire generated output. The goal of the LAQuer task is to provide the highlighted source spans

$s_1, \ldots, s_m$ that support the fact expressed in these highlights.

Within this setting, we aim that our LAQuer task definition would capture the following desiderata:

1) User-initiated Attribution Queries. Most attribution methods provide 'fixed', pre-determined attributions, meaning that attribution is generated alongside the output, only allowing users to explore it afterward (Gao et al., 2023b; Slobodkin et al., 2024; Phukan et al., 2024). However, we point out that users are often interested in checking the attribution only for a limited subset of facts within the generated output, and it is not possible to predict a user's specific interests in advance. LAQuer requires developing methods that can dynamically provide attribution for any arbitrary fact of interest, which the user highlights in the output.

2) Source and Output Localization. Slobodkin et al. (2024) introduce the Locally Attributable Text Generation task, where the goal is to provide the user with concise *source* spans necessary to verify a *complete* output sentence; in other words, the goal is to provide *localized*, precise input spans. In this work, we also consider the localization for the other side of the attribution, which is the *output* localization. Instead of complete output sentences, we work with output spans. Formally, the concatenation of the source spans should contain only the necessary content to support the information conveyed by the output spans.

3) Output Decontextualization. Given a contextualized claim c extracted from some text r, Choi et al. (2021) define a decontextualized claim m as one that uniquely specify entities, events, and other context such that the claim c is now interpretable. In our setting, it is likely that the highlights provided by the user are contextualized. For example, the output spans in Fig. 1 mention *"They,"* which refers to the consumers mentioned in the previous sentence. However, the user did not highlight *"consumers,"* because it is redundant and can be inferred from *"They."* Accordingly, source spans should correspond to a decontextualized version of the output. For example, in Fig. 1, the source from `nbcnews.com` must explicitly include *"consumers"* to avoid ambiguity. Only including *"they"* in the source spans would be problematic, as it lacks a clear referent and could lead to misinterpretation or false attributions. Formally, we denote the decontextualized meaning of the output spans in the context of the complete output as $I(o_1, \ldots, o_n|o)$. The source spans should express the decontextualized meaning of the output spans, $concat(s_1, \ldots, s_m) \models I(o_1, \ldots, o_n|o)$.

4 LAQuer Modeling Framework

The LAQuer setting, as defined above, inherently involves two processing stages, illustrated in Fig. 2. In the first stage, a source-grounded generation system generates a user-requested text, such as a summary or a long-form answer to a question, based on provided documents. This system may also include attribution metadata, mapping output segments to supporting source segments. For example, in Fig. 1, the generation system could output the sentence-level attribution underlined green. In our experiments (Section 5), we evaluate LAQuer using three generation methods: one without attributions and two recent attributed-generation approaches.

In the second stage, users who read the generated text can request localized attribution for specific facts by highlighting relevant spans. The LAQuer task then identifies the exact supporting source spans for the highlighted facts. Specifically, during stage 2, the LAQuer input consists of the following: (1) the source documents, based on which the output text was generated; (2) the generated output text; (3) the attribution metadata (if available); (4) the output spans highlighted by the user, which are assumed to correspond to a particular fact in the output text, for which attribution is sought.

Given these inputs, our proposed LAQuer method first performs a decontextualization step (A), which converts the input highlights into a coherent standalone sentence. Next, in the attribution step (B), we search for the supporting source spans that provide evidence for the decontextualized statement, where we explore two alternative methods for this step (prompt- and internals-based). This step leverages the attribution metadata from the generation step, if available, while also incorporating the extended highlights. These two steps are described in detail below.

4.1 Generating a Decontextualized Output Statement

As described in Section 3, a user's query consists of contextualized spans extracted from the output that depend on the surrounding text for full comprehension (e.g., the word *"consumers"* in Fig. 1). Step (A) of our method reformulates the selected spans

Highlighted output sentence	Decontexutalized Fact
The Los Angeles County Fire Department responded to multiple 911 **calls around 4:30 p.m.** at Penn Park, where the tree had toppled, trapping up to 20 people beneath its branches.	The 911 calls were made around 4:30.
The confirmation hearings for Supreme Court nominee Brett Kavanaugh . . . **Key issues included his views on presidential power**, abortion rights, and potential conflicts of interest regarding the Russia investigation.	Key issues included Brett Kavanaugh's views on presidential power.

Table 1: Examples illustrating our decontextualization step, drawn from Gunjal and Durrett (2024). Initially, LAQuer highlights (**bold**) are reformulated into decontextualized facts (→). These facts are subsequently aligned with revised highlights (←, underlined), to allow sentence-level attribution to incorporate additional context when needed. For example, in the second row, the mention of *"Brett Kavanaugh"* originates from a separate sentence, requiring the inclusion of additional source text to ensure accurate attribution.

into a self-contained *decontextualized* sentence, for which source attribution can be more easily sought in an unambiguous manner. We use the approach from Gunjal and Durrett (2024), as exemplified in Table 1.

This process may incorporate in the decontextualized statement additional phrases from the generated output text, beyond the user's initial highlights. For example, replacing the ambiguous "they" pronoun with the explicit "consumers" mention in Fig. 2, highlighted orange. Consequently, the obtained decontextualized statement includes all the information for which attribution should be identified within the source texts. If the query remains contextualized, this key information may be omitted, resulting in inaccurate attribution. By including the additional output span, the attribution used for the first sentence would be included, ensuring comprehensive coverage of the relevant content. For more details, see Appendix E.

4.2 Query-focused Attribution

Step (B) of our LAQuer method attributes the decontextualized sentence to the source texts, ensuring factual consistency while minimizing the retrieval of irrelevant spans. The effectiveness of this step depends on the generation method, particularly whether attribution metadata is available. Sentence-level attribution approaches, which provide fixed links between source and output spans, significantly reduce the search space, facilitating the localization of supporting evidence. In contrast, for non-attributed generation, the system must search the entire source document, increasing computational complexity.

For this step, we explore two approaches: one uses an LLM prompt while the other leverages the model's internal representations to identify alignments based on hidden state similarities (Phukan et al., 2024).

LLM-based Prompt Alignments. Leveraging the strong few-shot learning and reasoning capabilities of LLMs, we prompt an LLM to output the aligned spans. The prompt is listed in Fig. 4. Attributed source spans are separated by a semicolon (;). If a span does not match the source text, we apply a fuzzy search.[2] If the fuzzy search fails, we retry the prompt up to five times. If that also fails, we fall back to the original attribution provided by the attribution metadata, if available. Otherwise, we use all the source documents for attribution.

LLM-based Internals Alignments. Another strategy for achieving granular attribution is to compute the cosine similarity between the contextual hidden state representations of the source tokens and the output tokens (Dou and Neubig, 2021; Phukan et al., 2024). Phukan et al. (2024) has been shown to surpass GPT-4-based prompting methods in terms of accuracy, but was only evaluated in paragraph-level citations. In this work, we investigate its usefulness in LAQuer settings.[3] Compared to the previous LLM prompt-based approach, this method requires direct access to the model's weights, necessitating the use of open models.[4]

5 Experimental Setup

We evaluate the efficacy of our proposed framework in Section 4 by benchmarking multiple baseline methods for each stage in the process. We design an experimental setup that assesses both the quality of generated outputs and the accuracy of their

[2] https://github.com/google/diff-match-patch

[3] We re-implemented Phukan et al. (2024), as no source code was available.

[4] For more details on both approaches, see Appendix A.

Output sentence	Example decomposed fact
Exposing students to texts from different religions can be beneficial for their learning, as it helps them understand the development and advancement of societies, **promoting understanding**, respect, and fellowship.	Exposing students to texts from different religions promotes understanding.
Guns are rarely used in self-defense, are frequently stolen and used by criminals, and their **presence makes conflicts more likely to become violent**; armed civilians are unlikely to stop crimes and may make situations more deadly.	The presence of gun make conflict more likely to become violent.

Table 2: Example synthesized LAQuer inputs, simulating a user highlighting the output. First, output sentences are decomposed into atomic facts (→). Then, these facts are aligned back to highlights (←), denoted in **bold**.

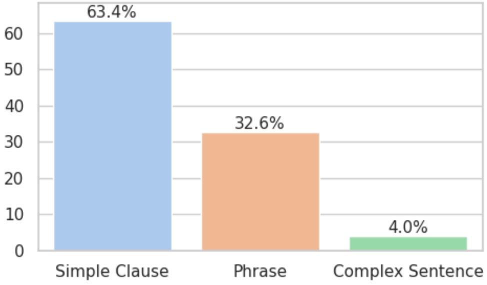


Figure 3: Distribution of span types based on syntactic complexity.

attributions. Our evaluation consists of automatic assessments on two key content-grounded generation tasks: Multi-Document Summarization (MDS) and Long-Form Question Answering (LFQA).

This section provides the foundation for benchmarking LAQuer and examining its effectiveness in reducing cognitive load while preserving factual consistency. We first introduce the datasets used in our experiments and describe the methodology for synthesizing attribution queries to simulate user fact-checking behavior (Section 5.1). Then, we describe our evaluation framework (Section 5.3), which measures the quality of the attribution under contextualized and decontextualized conditions.

5.1 Datasets

Our benchmark includes both a multi-document summarization setting (MDS) and a long-form QA setting setting (LFQA). Both are content-grounded settings such that source documents are used to generate an output. Specifically, we use SPARK (Ernst et al., 2024) for MDS,[5] and the RAG-based dataset curated by Liu et al. (2023) for LFQA.[6]

Synthesizing LAQuer Highlights for a Given Output. The source documents are used to generate outputs with attribution metadata, as described in Section 5.2. Given the outputs generated, we synthesize LAQuer inputs by simulating the user's process of highlighting relevant spans.

Our approach for generating highlights involves first decomposing each output sentence into atomic facts and then aligning these facts with the output sentence, exemplified in Table 2. To ensure our decomposition method closely mimics how users select contextualized spans, we adopt the contextualized decomposition approach from FActScore, which was specifically designed to break down long-form generations into atomic facts (Min et al., 2023). We use GPT-4o (OpenAI, 2024) for the decomposition. In order to align the generated output facts with the output, we use a naive lexical-based algorithm, described in Appendix E.

Our process for synthesizing facts results in an average of 2.6 facts per sentence. For each instance, we sample ten facts extracted from the entire output. We report the distribution of facts according to their syntactic complexity as a measure of how diverse the generated facts are. Specifically, we use the following categories:

1. **Phrase**: A span consisting of syntactic constituents without a complete clause structure (i.e., no finite verb or predicate). Example spans include: *"Kavanaugh past writings"*, *"A technical glitch"*.

2. **Simple Clause**: Contains at least one finite verb. Example spans: *"Judge Brett Kavanaugh faced intense scrutiny"*, *"His previous escape occurred in 2005"*.

3. **Complex Sentence**: Contains at least one embedded or subordinate clause and explicit discourse connectives. Example: *"Peach trees should be planted while they are dormant"*.

As illustrated in Fig. 3, the majority of extracted facts are simple clauses (roughly two-thirds), fol-

[5] SPARK is a subset of Multi-News (Fabbri et al., 2019)

[6] Statistics and more details are provided in Appendix C.

lowed by phrases (about one-third), with complex sentences making up only a small proportion.[7]

5.2 Generation Baselines

Following our suggested framework in Section 4, we benchmark three baseline methods for the first generation stage, from methods that provide no attribution to those that provide fine-grained attribution. Full details for the following methods are provided in Appendix B.

Vanilla. We include a naive baseline that generates text without attribution, as this represents a common approach in many real-world applications where attribution is not explicitly modeled. Evaluating this baseline allows us to measure the extent to which LAQuer methods can provide correct attribution on the entire source documents.

ALCE. Gao et al. (2023b) is a prominent attribution method that prompts the LLM to add citations at the end of each output sentence, in the form of square bracket, such as "... [1]." This method provides a fairly coarse-grained attribution, as citations point to an entire source document.

Attr. First. Slobodkin et al. (2024) divide the generation process into multiple explicit steps, allowing the attribution to be traced back to source spans. The first step, content selection, involves highlighting relevant source spans. The generation is then constrained to these selected spans, allowing the output to be tied back to the source. Unlike ALCE, which attributes at the document level, this approach attributes source spans, significantly reducing the costs associated with LAQuer while increasing the number of tokens required for generating the initial output. We analyze this trade-off in Section 6.3.

5.3 Evaluation

Our evaluation is comprised of different metrics for the quality of the output, following standard practices of each task, as well as the quality of the citations, adapted to the LAQuer setting. The purpose of measuring output quality is to show that overall methods that support localized attribution do not hurt output quality with respect to relevance. We incorporate both automated and human evaluations into our methodology.

Automatic evaluation To evaluate the quality of the output, we follow Slobodkin et al. (2024) and calculate Rouge-L (Lin, 2004) and BertScore (Zhang* et al., 2020), which were also used in their study. Additionally, we calculate METEOR (Banerjee and Lavie, 2005) and BLEURT20 (Sellam et al., 2020). Rouge-L and METEOR utilize n-gram comparisons, while BertScore and BLEURT20 are based on language models. All these metrics compare the generated output to a reference output. Lastly, we include a fluency metric based on MAUVE (Pillutla et al., 2021), which compares the distribution of the output to that of the reference texts.

To evaluate LAQuer citations, we sample ten facts from the facts extracted from the output, as described in Section 5.1. We then calculate AutoAIS (Gao et al., 2023a), which is an entailment metric commonly used for evaluating attribution. The metric outputs binary classification of whether an attributed source text supports an output fact, which is then averaged across all output facts to calculate the final score. Following Gao et al. (2023a), we make the distinction between evaluating entailment with contextualized facts and decontextualized facts. We source contextualized facts from the process described in Section 5.1, and decontextualized facts from the process described in Section 4.2.

Additionally, we measure the attributed text length in content words[8] to confirm that our method significantly reduces unnecessary reading. Lastly, we report the percent of non-attributed facts.[9]

6 Results and Analyses

6.1 Main Results

The output quality metrics are reported in Table 3. ATTR. FIRST outperforms other methods in terms of ROUGE-L, METEOR and MAUVE, while the Vanilla generation outperforms in terms of BLEURT-20. This suggests that ATTR. FIRST has more lexical overlap with the reference outputs, while the Vanilla generation is more semantically similar. In general, both methods achieve similar output quality results, with ALCE lagging behind.

The citations quality metrics are reported in Table 4. We make the following observations.

LAQuer methods significantly and attractively reduce the length of the attributed text. Across

[7] Categorization was performed using the SpaCy NLP toolkit: `https://spacy.io/`

[8] Excluding stop-words `https://nltk.org`

[9] More details are provided in Appendix C.2.

	Method	R-L ↑	METEOR ↑	BERTSCORE ↑	BLEURT-20 ↑	MAUVE ↑
MDS	VANILLA	19.2 ±0.6	28.3	86.4 ±0.2	**43.0** ±0.7	59.8
	ALCE	19.4 ±0.6	27.3	86.1 ±0.2	38.2 ±0.8	63.7
	ATTR. FIRST	**21.1** ±0.7	**29.7**	**86.6** ±0.2	41.1 ±0.9	**84.9**
LFQA	VANILLA	37.2 ±3.2	45.6	**90.7** ±0.6	**60.5** ±1.7	81.5
	ALCE	34.4 ±2.7	44.3	90.1 ±0.5	56.8 ±1.7	90.6
	ATTR. FIRST	**38.2** ±2.7	**46.1**	90.6 ±0.6	58.5 ±1.8	**96.7**

Table 3: Generated text quality results, averages include standard error of the mean.

		Method	AUTOAIS CON. ↑	AUTOAIS DECON. ↑	LENGTH ↓	NON ATT. (%) ↓
MDS		VANILLA	**82.2** ±1.6	**84.5** ±2.0	1681.6 ±205.5	0.0
	🟨	LLM Prompt	62.5 ±2.0	49.7 ±2.5	32.0 ±1.8	0.0
		LLM Internals	18.0 ±1.7	13.1 ±1.5	28.1 ±0.9	0.0
		ALCE	67.4 ±2.3	74.8 ±2.3	979.1 ±117.8	5.2 ±0.8
	🟨	LLM Prompt	55.8 ±2.2	44.3 ±2.4	41.6 ±3.4	5.2 ±0.8
		LLM Internals	15.5 ±1.6	10.2 ±1.5	29.9 ±8.2	8.2 ±1.2
		ATTR. FIRST	80.3 ±2.2	58.0 ±2.8	33.0 ±2.4	0.4 ±0.2
	🟨	LLM Prompt	71.5 ±2.3	42.4 ±2.4	14.6 ±0.5	0.4 ±0.2
		LLM Internals	28.6 ±2.4	13.2 ±1.7	**12.2** ±0.4	21.4 ±0.9
LFQA		VANILLA	69.5 ±4.6	71.0 ±4.5	4636.8 ±488.3	0.0
	🟨	LLM Prompt	65.1 ±3.8	65.4 ±4.3	38.1 ±2.6	0.0
		LLM Internals	19.0 ±2.8	18.0 ±2.4	24.9 ±1.5	0.0
		ALCE	50.8 ±4.8	55.6 ±5.1	2346.0 ±300.2	13.8 ±4.2
	🟨	LLM Prompt	56.8 ±4.0	52.8 ±3.9	42.0 ±10.9	13.8 ±4.2
		LLM Internals	13.0 ±2.4	12.8 ±2.4	26.6 ±1.5	17.1 ±3.0
		ATTR. FIRST	**88.0** ±3.4	**83.9** ±3.3	43.3 ±2.4	0.0
	🟨	LLM Prompt	83.0 ±3.1	69.6 ±4.3	17.3 ±0.8	0.0
		LLM Internals	46.6 ±4.0	37.8 ±4.4	**14.3** ±0.7	7.0 ±1.7

Table 4: LAQuer citation results, averages include standard error of the mean. We separately calculate AutoAIS for contextualizd (Con.) and decontextualized (Decon.) output facts. 🟨 indicates LAQuer methods and yellow indicates the best LAQuer method. Non Attributed measures the percentage of facts without attribution.

		Method	AIS DECON. ↑
MDS		VANILLA	**91.5** ±2.3
	🟨	LLM Prompt	39.0 ±4.2
		ATTR. FIRST	54.3 ±4.4
	🟨	LLM Prompt	31.6 ±3.8
LFQA		VANILLA	**90.9** ±5.1
	🟨	LLM Prompt	59.5 ±7.7
		ATTR. FIRST	53.6 ±8.6
	🟨	LLM Prompt	50.2 ±9.2

Table 5: LAQuer human evaluation of citation results, averages include standard error of the mean. 🟨 indicates LAQuer methods.

all methods, LAQuer reduces attribution length by two orders of magnitude for Vanilla and ALCE, and by an average of 59% for ATTR. FIRST. For example, in the Vanilla setting, which does not rely on a particular generation method but does not provide any attribution, LAQuer attribution can direct the user to correct highly localized supporting spans in nearly two thirds of the cases.

The LLM prompt is the best-performing LAQuer method. In all generation methods, we find that the LLM prompt performs the best in terms of AutoAIS, significantly surpassing the LLM internals method. This is true for both MDS and LFQA settings. The LLM internals method has low performance across all generation methods. The best results for the LLM internals are achieved when the source is localized with ATTR. FIRST, suggesting that it struggles with localization of document-level texts. In addition, when the LLM internals method is applied on top of source-localized attribution methods, ALCE and ATTR. FIRST, we observe an increase in non-attributed output words.

LAQuer methods can leverage localized attributions provided by ATTR. FIRST. Even without applying LAQuer, ATTR. FIRST provides very concise sentence-level attribution, averaging only 36 characters. This means that the localized support for the LAQuer fact needs to be identified only within a quite short span. Consequently, the strong performance of ATTR. FIRST carries over to LAQuer. In the contextualized setting, ATTR. FIRST is the top-performing LAQuer method, indicating that LAQuer methods can leverage initially localized attributions provided by the generation method itself. However, in the decontextualized setting, ATTR. FIRST yields notably low AutoAIS scores, as low as 58 for MDS, and 53.6 for LFQA in our manual evaluation, described in Section 6.2. These low scores limit the effectiveness of LAQuer methods, since the necessary evidence for the decontextualized facts is absent from the original ATTR. FIRST provided spans. We hypothesize that this degredation stems from ATTR. FIRST failure to decontextualize its attributions. This suggests that when generating attributions for localized output segments, it is crucial to first decontextualize these output spans, and accordingly to make sure to support also the decontextualizing information within the source attributions.

6.2 Human Analysis

To further assess our findings, we report a small-scale human annotation conducted by the authors using our most promising methods. We annotated 20 examples per task, each for the Vanilla and ATTR. FIRST methods, both with and without LAQuer, resulting in 80 examples per task (160 in total). For each example, we calculate AIS (Rashkin et al., 2023) at the decontextualized fact-level. For AIS, similar to the AutoAIS metric, the annotator is asked to make a binary classification of whether an output fact is supported by the attributed source texts; we then average classifications across all output facts to calculate the final score.

Our results are reported in Table 5. In accordance with our main results in Table 4, we find that LAQuer methods struggle with decontextualized facts. From this analysis, we observe that the model often omits the document's broader theme. For example, in Table 11, the LLM prompt method correctly attributes multiple "issues", yet it fails to attribute "Supreme Court".

6.3 Cost Analysis

We provide the average size of prompts in Table 7. On one hand, we find that LAQuer prompts in ATTR. FIRST are an order of magnitude smaller than in Vanilla generation. On the other hand, ATTR. FIRST generation is costly, inducing an increase of 90% in prompt length compared to Vanilla generation, as reported by Slobodkin et al. (2024). These results suggest that increased computational cost during generation can lead to more efficient LAQuer methods.

6.4 Estimate for LAQuer Localization

To better understand the potential benefits of LAQuer, we estimate the average amount of text required to support an output fact or sentence. We compare this across different levels of source granularity, including source spans, source sentences, and entire source documents. For this analysis, we utilize the SPARK dataset (Ernst et al., 2024), which is used in our study and contains fine-grained, human-annotated attribution.

Source granularity	Output facts	Output sentences
Spans	128.0	231.4
Sentence	278.5	485.1
Document	4679.5	7226.6

Table 6: Analysis of attribution lengths (measured in characters) with varying granularities, based on the SPARK dataset (Ernst et al., 2024).

Our analysis, summarized in Table 6, presents the average number of characters to read under different attribution granularities. LAQuer operates at both the source and output fact levels, requiring an average of 128 characters to read. In contrast, ATTR. FIRST attributes at the output sentence level with source spans, resulting in an average of 278.5 characters. This finding highlights the benefits of localizing attribution per output fact, reducing the text users need to read by 54%.

7 Conclusion

In this work, we introduce a novel motivation for post-hoc attributed text generation, enabling users to create localized attribution queries, LAQuer. We introduce a challenging benchmark, which subsumes existing attribution methods by considering both the generation and post-hoc steps. Our results show that LAQuer methods significantly reduce attribution length, but LAQuer attribution remains a challenging task for decontextualized facts. In addition, our methods are associated with a high cost of LLM calls, suggesting future research should focus on creating more efficient frameworks. Lastly, there is a performance gap between different generation methods.

Limitations

Addressing attribution queries increases computational cost on top of fixed sentence-level or token-level attribution. In Section 6.3, we discuss the trade-off between computational cost during generation and that during attribution.

While our work is focused on content-grounded generation, LAQuer could be applied to outputs generated by the model's parametric knowledge, by retrieving the documents after the generation rather than before. We leave such exploration for future work.

AutoAIS is used as a key metric for evaluating attribution quality, which is an LLM-based automated metric. We conducted a small-scale human analysis to support these results in Section 6.2, finding similar trends.

Ethical Considerations

The ability to attribute outputs of LLMs to specific sources is crucial for transparency, accountability, and trust in AI-generated content. Our work contributes to this goal by simplifying the attribution process for users and making it more localized. However, errors in attribution can mislead users into assuming a stronger or weaker connection between the generated content and its source than what actually exists.

We utilized AI-assisted writing tools during the preparation of this paper to improve clarity and coherence. However, all content was carefully reviewed and edited by the authors to ensure accuracy.

Acknowledgements

We would like to thank our reviewers for their constructive suggestions and comments. This work was supported by the Israel Science Foundation (grant no. 2827/21), NSF-CAREER Award 1846185, and NSF-AI Engage Institute DRL-2112635.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Lierni Sestorain Saralegui, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. 2023. Attributed question answering: Evaluation and modeling for attributed large language models. *Preprint*, arXiv:2212.08037.

Jan Buchmann, Xiao Liu, and Iryna Gurevych. 2024. Attribute or abstain: Large language models as long document assistants. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8113–8140, Miami, Florida, USA. Association for Computational Linguistics.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. 2024. Contextcite: Attributing model generation to context. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Qiang Ding, Lvzhou Luo, Yixuan Cao, and Ping Luo. 2024. Attention with dependency parsing augmentation for fine-grained attribution. *Preprint*, arXiv:2412.11404.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Ori Ernst, Ori Shapira, Aviv Slobodkin, Sharon Adar, Mohit Bansal, Jacob Goldberger, Ran Levy, and Ido Dagan. 2024. The power of summary-source alignments. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6527–6548, Bangkok, Thailand. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Anisha Gunjal and Greg Durrett. 2024. Molecular facts: Desiderata for decontextualization in LLM fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3751–3768, Miami, Florida, USA. Association for Computational Linguistics.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. 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. Association for Computational Linguistics.

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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

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

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes. *ArXiv*, abs/2203.11147.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *ArXiv*, abs/2401.06855.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Anirudh Phukan, Shwetha Somasundaram, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. 2024. Peering into the mind of language models: An approach for attribution in contextual question answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11481–11495, Bangkok, Thailand. Association for Computational Linguistics.

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

Jirui Qi, Gabriele Sarti, Raquel Fernández, and Arianna Bisazza. 2024. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6037–6053, Miami, Florida, USA. Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840.

Tal Schuster, Adam Lelkes, Haitian Sun, Jai Gupta, Jonathan Berant, William Cohen, and Donald Metzler. 2024. SEMQA: Semi-extractive multi-source question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1363–1381, Mexico City, Mexico. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. 2024. Attribute first, then generate: Locally-attributable grounded text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3344, Bangkok, Thailand. Association for Computational Linguistics.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam M. Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, Yaguang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping

	Method	Input Length	Output Length
MDS	Vanilla	25674.7 ±396.7	227.7 ±6.6
	ALCE	22239.6 ±279.5	214.5 ±6.8
	Attr. First	2843.0 ±7.3	89.3 ±2.0
LFQA	Vanilla	58299.8 ±1031.1	232.4 ±8.0
	ALCE	45104.4 ±826.6	200.9 ±8.1
	Attr. First	3025.4 ±7.7	107.2 ±3.4

Table 7: Average number of characters in the LLM prompt LAQuer method, including standard error of the mean.

Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, I. A. Krivokon, Willard James Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Hartz Søraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Díaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, V. O. Kuzmina, Joseph Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Rogers Croak, Ed H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *ArXiv*, abs/2201.08239.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A LAQuer Methods Details

In this section, we provide a full description of the LAQuer methods used, described in Section 4.

A.1 LLM Prompt

The prompt is provided in Fig. 4. The average size of prompts is reported in Table 7. We use GPT-4o (OpenAI, 2024). In our experiments, we include three in-context examples sourced from the dev split of the corresponding datasets. We manually optimized the prompt instructions and few-shot examples based on iterations on the development set.

A.2 LLM Internals

Our LLM-based internals method is based on the method by Phukan et al. (2024). Since this method requires access to the weights of the model, we run Llama-3.1-8B-Instruct on a single A100-80GB GPU for approximately 8 hours. More running time details are available in Table 8.

```
You are provided with an output sentence and
the source texts from which it was generated.
You need to identify spans in the source
from which the output sentence was generated.
Copy verbatim the attributing source spans,
and use a semicolon (;) as a delimiter
between each consecutive span. The output
sentence should be fully supported by the
concatenation of the attributed source spans.
IMPORTANT: Each span must be verbatim copied
from the corresponding sources. Do not make
any changes or paraphrases to the source spans.
If necessary, you may copy multiple spans
from the same or source, but avoid adding
un-necessary spans and keep each span as short
as possible.

Input:
Source 1: Voters in 11 states will pick their
governors tonight Republicans appear on track
to increase their numbers by at least one,
and with the potential to extend their hold
to more than two-thirds of the nation's top
state offices
...

Output: There is a race for the governor's
mansion in 11 states today.

Attribution: Voters in 11 states will pick
their governors tonight
```

Figure 4: Example prompt for LLM-based post-hoc alignment. The instructions are depicted in green, input to the model in black, and model's output in red. This example is one of three few-shot examples. The source texts of the few-shot examples are adapated based on the generation method: Vanilla includes all documents, ALCE includes relevant documents, and Attr. First includes relevant source spans.

We now provide a short description of this work, and the adaptation we made to support the LAQuer setting. The method proposed by Phukan et al. (2024) is based on the idea that LLMs have inherent awareness of the document parts they use while generating answers. They claim that it is likely captured by the hidden states of the LLM. Accordingly, their method includes creating a prompt that concatenates the query q, the documents D, and the output o, and then feeds this to a LLM in a single forward pass. This creates the hidden representations of the text.

Formally, the prompt is denoted P, such that $P = q + D + o$, where '+' denotes concatenation. Also, the hidden layer representation of token $t_i \in P$ for layer l of the model is denoted h_i^l. The attribution process is then composed of two sub-tasks:

Sub-Task 1: Identification of extractive answer tokens An important claim made in their paper

	Method	Avg. time (sec.)
MDS	Vanilla	0.6 ±0.0
	ALCE	4.2 ±0.2
	Attr. First	8.6 ±0.6
LFQA	Vanilla	0.7 ±0.0
	ALCE	20.1 ±2.1
	Attr. First	54.1 ±4.0

Table 8: Average time of the LLM internals LAQuer method, including standard error of the mean.

is that not all tokens should be attributed, because some tokens are 'glue' tokens created by the LLM. This task involves identifying extractive tokens, which are tokens that originate from the source documents, usually verbatim.

Formally, a token $o_i \in o$ is an extractive token if there exists a token $d_j \in D$ such that the cosine similarity between h_i^l and h_j^l is greater than a threshold θ.

In our work, we use the threshold $\theta = 0.7$ and layer $l = 5$, which achieves the highest F1 scores based on their paper. In addition, as formalized in Section 3, we only look at output spans $o_1, \dots, o_n$ provided as input, and not the entire output o.

Sub-Task 2: Attribution of extractive answer span S Given an output span S with tokens $o_1, \dots, o_m \subseteq o$, compute the average hidden layer representation h_S for each token $o_i \in S$ as:

$$h_s = \frac{1}{n} \sum_{i=1}^{n} h_i^l$$

Next, h_s is used to identify anchor tokens in D. Anchor tokens, denoted D_T, are the tokens most similar to the output span S. This is calculated for each document token $d_j \in D$ as the cosine similarity between h_S and h_j^l. For each anchor token $d_a \in D_T$, a window of tokens around d_a is explored, up to a length L. For each window, an average representation is calculated and the highest ranked window is considered the attribution for S. In our work, we use $L = 30$.

B Generation Methods Details

In this section, we provide a full description of the generation methods used, described in Section 4.

As as a pre-processing step, we first decontextualize the output spans. We use the decontextualization prompt from MolecularFacts (Gunjal and Durrett, 2024), which takes the concatenated output spans as input, together with the entire output as context, and outputs decontextualized facts. We used the original MolecularFacts prompt and ran it with GPT-4o. The resultant decontextualized fact is then mapped back to the output, as described in Appendix E.

B.1 ALCE

Gao et al. (2023b) introduced the idea of allowing LLMs to generate citations together with the output. We use the same prompt as the original paper with two few-shot examples and $T = 0.5$, following Slobodkin et al. (2024).

B.2 Attr. First

Slobodkin et al. (2024) decompose the generation process into multiple explicit steps, allowing for precise attribution tracing. The first step, content selection, involves highlighting relevant source spans. The second step, sentence planning, consists of clustering spans for each sentence, followed by sentence generation based on the clustered information. Each new sentence is generated with conditioning on the previously generated sentences. We adopt the same prompt and few-shot demonstration examples as used in the original paper. Among the multiple variants of Attr. First, we select Attr. First$_{CoT}$, which the paper identifies as the best-performing variant.

C Experimental Setup Details

C.1 Datasets

Our benchmark includes both a multi-document summarization setting (MDS), as well as a long-form QA setting (LFQA). Both are content-grounded settings such that the source texts are used to generate an ouput. Specifically, we use SPARK (Ernst et al., 2024) for MDS, and the RAG-based dataset curated by Liu et al. (2023) for LFQA. We used the same split of the datasets created by Slobodkin et al. (2024). The datasets sizes are provided in Table 9. Both datasets are in English. The licenses for the datasets are following: Ernst et al. (2024) CC BY-SA 4.0, Liu et al. (2023) MIT license.

C.1.1 Synthesizing Attribution Queries

Following Section 5.1, we provide more details about the decomposition of an output text to output facts. We first split the output into sentences.[10]

[10] using spaCy https://spacy.io/

Task	Dataset	Dev	Test
MDS	SPARK (Ernst et al., 2024)	45	65
LFQA	Evaluating (Liu et al., 2023)	44	45

Table 9: Datasets sizes used in our benchmark for development and evaluation.

For each output sentence, we then run a prompt decomposing the output into atomic facts. FActScore (Min et al., 2023) is an LLM-based method used to breakdown a sentence into atomic facts. It is a prompt comprised of instructions and multiple few-shot examples. We used the original FActScore prompt and run it with GPT-4o. The resultant fact is then mapped back to the output, as described in Appendix E.

C.2 Evaluation

For calculating AutoAIS, we use the model GOOGLE/T5_XXL_TRUE_NLI_MIXTURE (Honovich et al., 2022), which is trained on NLI datasets and has been used in previous work to analyze attribution (Gao et al., 2023a; Slobodkin et al., 2024). It correlates well with AIS scores (Gao et al., 2023a).

D Attribution Metadata Details

Illustrated in Fig. 2, we suggest that some generation methods can provide attribution metadata. In this section, we discuss the attribution metadata provided by the Attr. First method. Each sentence-level localized attribution is composed of one or more records, each consisting of the following information: output sentence idx, source file ID, and a list of source character offsets. For example, '<0, doc_1.txt, [[17367, 17562]]>'. In comparison to non-localized attribution, such as the ALCE method, this requires one additional column for offsets. We analyze the average storage required for saving sentence-level attribution per output. Our analyses show that it requires 2Kb on average per attributed output, totalling in a fairly small increase of 700 bytes per attributed output.

E Alignment of Facts to Spans

Throughout our work, we extracted facts from the output text and later needed to map them back to their corresponding spans. In this section, we describe the algorithm used to align extracted facts with the original output text.

The first application of this alignment process is in our evaluation methodology, where we decompose each output sentence into atomic facts using an LLM, as detailed in Section 5.1. For instance, consider the sentence "Exposing students to texts from different religions promotes understanding" from Table 2. To simulate a user's highlight, we need to align these atomic facts with spans in the output, providing the necessary spans for the LAQuer method. In this example, the aligned highlight would be "exposing students to texts from different religions ... promoting understanding."

The second application is in our proposed method, where we decontextualize queries. For example, in Table 1, we need to align the fact "The 911 calls were made around 4:30" with the output text "The ... 911 calls around 4:30 p.m." This alignment is crucial to ensure proper attribution, such as correctly highlighting the word "911."

To achieve this alignment, we implement a naive lexical alignment algorithm. This approach is expected to perform well since each output fact is extracted from a single output sentence, and the generated fact does not contain any paraphrases.

Formally, given an output o and a fact f expressed by o, we wish to find spans $o_1, \ldots, o_n \subseteq o$ such that $f \models concat(o_1, \ldots, o_n)$.

Alignment algorithm

1. **Tokenization & Lemmatization:** We first split the output o into words $o_1, \ldots, o_n$, and the fact f into words $f_1, \ldots, f_m$. Each word is lemmatized.[11]

2. **Edit Script Calculation:** We compute the edit script[12] between the output words and the fact words. The edit script represents the minimal set of operations (insertions, deletions, and substitutions) required to transform one sequence into the other. Each word in the output is assigned an edit operation.

3. **Word Alignment Based on Edit Operations:** Any output word o_i classified as unchanged is considered aligned to the corresponding fact word f_j.

The advantage of using an edit script is that it considers the order in which the words appeared. However, sometimes the fact transposes information from the output sentence. For example, in

[11] using spaCy https://spacy.io/

[12] Using Levenshtein distance https://nltk.org/

the second row of Table 2, the fact mentions "public school" after the mention of "the First Amendment", but in the output sentence the order is reversed. The algorithm will then not be able to align "public school". To support such transpositions, we generate a new fact f' with non-aligned words from f. We then run this algorithm recursively with f'.

Overall, in 88% of the examples we are able to align all content words,[13] and in 99% we are able to align all content words but one.

[13] Excluding stop-words `https://nltk.org`

	Example
Output sentence	The confirmation hearings for Brett Kavanaugh were marked by controversy over the withholding of documents, with Democrats repeatedly complaining that Republicans and the White House were keeping important records from the public and the committee.
LLM Prompt	Such theatrics have characterized Kavanaugh’s hearings, in which **Democrats have repeatedly complained that Republicans have withheld documents from the committee and the public that shed important light on Kavanaugh’s past**. . . . Democrats have repeatedly complained that the White House is withholding tens of thousands of documents relevant to the nomination and wants many more that have been provided released to the public.
LLM Internals	Such theatrics have characterized Kavanaugh’s hearings, in which **Democrats have repeatedly complained that Republicans have withheld documents from the committee and the public that shed important light on Kavanaugh’s past**.

Table 10: Example MDS result. Top: one example output sentence from the ATTR. FIRST baseline with synthesized LAQuer highlights. Bottom: the predicted attributions, with correct attribution in **bold**.

	Example
Output sentence	The upcoming Supreme Court term is poised to address several contentious issues that could significantly impact American society and politics.
LLM prompt	After a year in which liberals scored impressive, high-profile **Supreme Court** victories, conservatives could be in line for wins on some of **this term’s most contentious issues**, as the justices consider cases that could gut public sector labor unions and roll back affirmative action at state universities. . . . A potential body blow to labor Public-employee unions and politicians of both parties are keenly focused on a California dispute about whether states can compel government employees to pay union dues. . . . Higher ed affirmative action back in the crosshairs . . . The meaning of "one person, one vote’ . . . Testing when abortion clinic regulations go too far . . . The death penalty is shaping up to be a big issue for the Supreme Court as it begins a new term
LLM internals	However, as the **court’s new term kicks off** Monday, uncertainty surrounds several other politically potent cases that could wind up on the court’s agenda. Litigation over state efforts to limit abortion by regulating clinics and doctors is making its way to the high court. Lois Lerner should have been gone shortly after the scandal first unraveled.

Table 11: Example MDS result. Top: one example output sentence from the Vanilla baseline with synthesized LAQuer highlights. Bottom: the predicted attribution, with correct attribution in **bold**.