ALiiCE: Evaluating Positional Fine-grained Citation Generation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) can enhance the credibility and verifiability by generating text with citations. However, existing tasks and evaluation methods are predominantly limited to sentence-level statement, neglecting the significance of positional fine-grained citations that can appear anywhere within sentences. To facilitate further exploration of the fine-grained citation generation, we propose ALiiCE, the first automatic evaluation framework for this task. Our framework first parses the sentence claim into atomic claims via dependency analysis and then calculates citation quality at the atomic claim level. ALiiCE introduces three novel metrics for positional fined-grained citation quality assessment, including positional fine-grained citation recall and precision, and coefficient of variation of citation positions. We evaluate the positional fine-grained citation generation performance of several LLMs on two long-form QA datasets. Our experiments and analyses demonstrate the effectiveness and reasonableness of ALiiCE. The results also indicate that existing LLMs still struggle to provide positional fine-grained citations.

1 Introduction

Large Language Models (LLMs; Brown et al., 2020) can improve performance in several NLP tasks by incorporating external knowledge (Lewis et al., 2020). In order to improve LLMs' credibility, Gao et al. (2023b); Liu et al. (2023) propose a new paradigm for long-form QA, which LLMs are required to provide citations to the retrieved passages for the statements they generate. Since then, many studies (Ye et al., 2024; Huang et al., 2024; Slobodkin et al., 2024) have focused on how to enhance LLMs' citation generation capabilities.

However, existing methods and evaluation metrics on citation generation are predominantly limited to sentence-level statements. Malaviya et al. (2024) suggest that a sentence might not be the smallest unit capable of representing an atomic claim, potentially leading to inaccurate evaluations. As illustrated in Figure 1, the response A1 actually contains two different claims, but the sentence-level citation treats the entire sentence as one claim. Additionally, Liu et al. (2023) highlight that the generated text scope of a single in-line citation is often ambiguous. Citations of A1 in Figure 1 is ambiguous, because the citation marks at the end of A1 do not clearly indicate whether they support both claims or only the last claim.

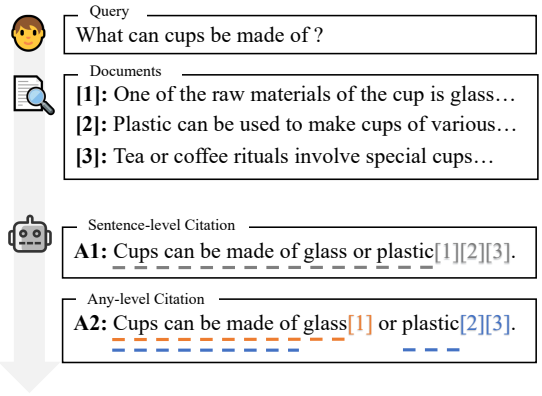

Figure 1: "Sentence-level" vs. "Any-level" in the task of citation text generation. The text with grey underline corresponds to the claim in A1 cited by "[1][2][3]". The texts of orange and blue underlines correspond to the claims in A2 cited by "[1]" and "[2][3]", respectively.

In fact, in many long-form contexts, particularly in professional fields such as academic writing (Funkquist et al., 2023), citation marks often appear in the middle of a sentence rather than always at the end, as response A2 shown in Figure 1. Compared with sentence-level citation, the advantages of this fine-grained generation are: 1) clearer indication of the text scope associated with each citation mark, and 2) better user-friendliness, allowing users to locate more specific content to check. We refer to this improved generation task as Positional Fine-grained Citation Text Generation.

Despite the importance of this task, an effective evaluation method has yet to be developed. Sentence-level metrics simply merge citations from different positions, treating the entire sentence as a single claim (Gao et al., 2023b; Yue et al., 2023). In this case, when using Natural Language Inference (NLI; Honovich et al., 2022) to judge whether the claim is supported by its evidences, sentence-level metrics can easily result in the issue of excessively long NLI contexts when multiple atomic claims occur simultaneously within the sentence. Furthermore, if there is an overlap between evidence of different atomic claims, sentence-level judgments can also become unreasonable, for correct citations might be mistakenly excluded.

To effectively assess fine-grained citations at atomic claim level, we propose **ALiiCE**, representing **A**utomatic **LLM**'s Pos**i**tional F**i**ne-grained **C**itation **E**valuation. Our method first employs a Dependency Tree based approach to parse the atomic claim corresponding to each citation in the response. For instance, the two claims of sentence A2 in Figure 1 are parsed as "Cups can be made of glass" and "Cups can be made of plastic". Further, our method incorporates three new metrics on citation quality, including positional fine-grained citation recall and precision, as well as coefficient of variation of citation positions for assessing the dispersion of citation placements within a sentence.

We conduct experiments to evaluate the performance of existing models on positional fine-grained citation generation. We employ two long-form QA datasets, ASQA (Stelmakh et al., 2022) and ELI5 (Fan et al., 2019) to evaluate LLMs including GPT-3.5, GPT-4 and LLaMA-3-8B.

Through the analyses on experimental results and cases, we demonstrate that ALiiCE is an effective and reasonable evaluation method for positional fine-grained citation generation. We observe that existing LLMs face challenges in generating positional fine-grained citations. These challenges encompass producing a limited number of fine-grained citations and struggling to generate accurate ones. In addition, we find that the latest open-source LLMs narrow the gap in citation generation with closed-source LLMs. We hope that our work can inspire more research into positional fine-grained citation text generation.

In summary, our main contributions include:

- We propose a dedicated evaluation method for positional fine-grained citation generation.
- We analyze and verify the shortcomings of sentence-level evaluation in positional fine-grained citation text generation, as well as the effectiveness of our method.
- We evaluate the performance of existing LLMs on positional fine-grained citation text generation on long-form QA datasets.

2 Background & Task Definition

In this section, we briefly introduce the background of our research and provide a definition of positional fine-grained citation generation.

2.1 Citation Generation in Long-form QA

Long-form QA is a type of Question-Answering task, where the answer to a question is detailed, comprehensive, and typically longer than brief answers. Unlike short-form QA, which typically provide binary, entity-level or short sentence answers, long-form QA generates more elaborate responses that include explanations, context, and additional relevant information (Krishna et al., 2021).

Citation generation involves producing citation marks (namely, document IDs) while generating text, indicating the source documents on which the text is based (Funkquist et al., 2023). In our work, we focus on citation generation for long-form QA. Unlike traditional task, positional fine-grained citation generation allow citation marks to appear at any position within the sentence.

2.2 Task Definition

Formally, given a query q and a set $\mathcal{D}$ of retrieved passages based on q, the generator $\mathcal{M}$ is required to generate a response $\mathcal{R}$ consisting of n sentences $s_1, \ldots, s_n$. We assume that the j-th item in i-th sentence s_i is $s_{i,j}$, which has two situations:

$$s_{i,j} = \begin{cases} \mathcal{C}_{i,j}, & \text{if } s_{i,j} \text{ is a group of citation marks} \\ x_{i,j}, & \text{if } s_{i,j} \text{ is a word} \end{cases} \tag{1}$$

where $\mathcal{C}_{i,j} = \{c_{i,j,1}, c_{i,j,2}, \ldots\}, c_{i,j,k} \in \mathcal{D}$. If a $s_{i,j}$ is a group of citation marks, then it has its corresponding claim generated by $\mathcal{M}$ based on $\mathcal{C}_{i,j}$, denoted as $\mathcal{A}_{i,j}$. Obviously, $\mathcal{A}_{i,j}$ is constructed from $\{x_{i,j}\}$, as a part of sentence s_i.

Take A2 in Figure 1 as an example, "plastic" is a word, and "[2][3]" is a group of citation marks with the claim "Cups can be made of plastic".

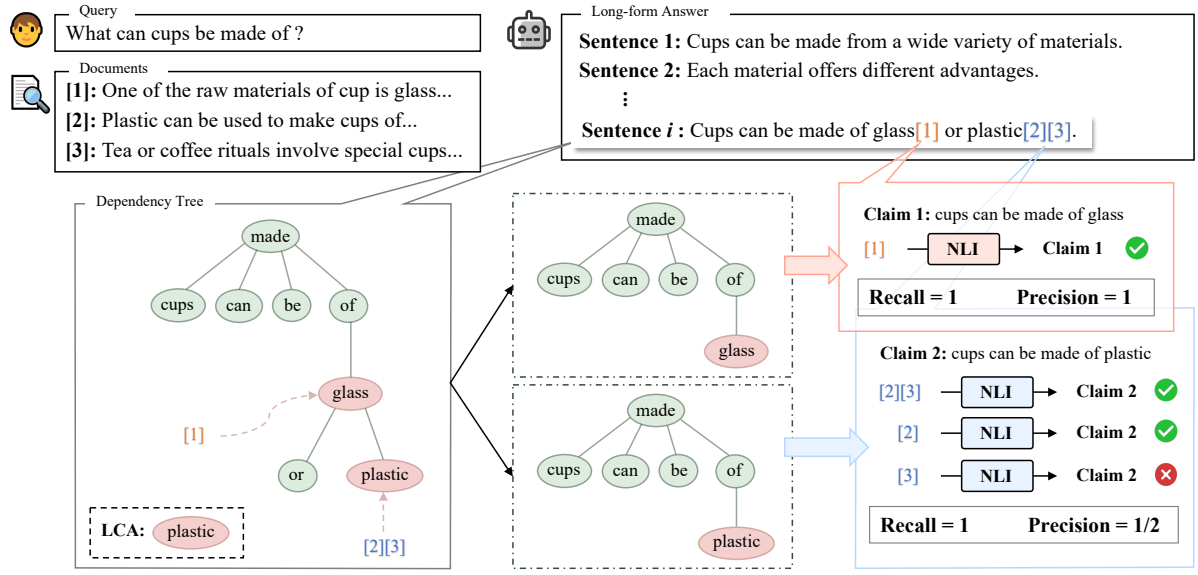

Figure 2: An example of ALiiCE evaluation framework on positional fine-grained citation generation. Given a query and related documents, the LLM generate a long-form answer. For sentence i in answer, the parsing pipeline involves constructing the dependency tree, identifying the LCA node to obtain the modified tree of each claim, and converting modified trees into texts. Finally, we calculate the citation recall and precision for each claim.

3 ALiiCE: Automatic LLMs' positional fine-grained Citation Evaluation

In this section, we give a detailed description of ALiiCE. First, we introduce how we construct the atomic claim parsing pipeline based on dependency trees and Lowest Common Ancestor (LCA). Then, we present our novel metrics for the evaluation of positional fine-grained citation quality.

3.1 Dependency Tree

Dependency tree is a hierarchical representation of the grammatical structure of a sentence, showing how words rely on each other (Culotta and Sorensen, 2004). Compared with the hierarchical syntax tree based on operators, the dependency tree is more concise, making it easier to extract the relationship between sentence components. And in a dependency tree, a subtree can represent a phrase or clause that depends on its root. Thus, dependency tree is highly suitable for extracting the claims associated with the citation marks.

In ALiiCE, we employ dependency trees to describe sentences in $\mathcal{R}$ for subsequent parsing stage. In practice, for every $\mathcal{C}_{i,j}$, we match a node of word $x_{i,j}$ in the tree of sentence s_i, so that every node in dependency tree is a word.

3.2 Lowest Common Ancestor

Lowest Common Ancestor (LCA) is the lowest (i.e. deepest) node of two different nodes possessing both of them as descendants in a tree. When there are multiple citations in different locations within a sentence, their respective claims need to be parsed. To parse a claim of $\mathcal{C}_{i,j}$, we need to exclude irrelevant content from other claims, as different claims may share identical sentence components. In the dependency tree, for two distinct nodes, their dependency traces back to their LCA node. The subtrees of the LCA node's children, containing two word nodes matched by two citations, represent two components that depend on the LCA node. Based on the relationship between two subtrees, we can delete or modify one of them, thereby achieving claim extraction of the citation of the other subtree.

3.3 Parsing Pipeline

Our parsing pipeline is illustrated by Figure 2 and simplified pseudo code is shown in Algorithm 1.

For a sentence s_i in response $\mathcal{R}$, we extract groups of citation marks $\{\mathcal{C}_{i,j}\}$ from difference positions. Then we do text cleaning on s_i to obtain raw sentence s_i', involving removing citation marks and other punctuation. s_i' is used to construct dependency tree T. Next, for each $\mathcal{C}_{i,j}$, we match a word node, which we denote as a citation node, for simplicity. The principle of matching nodes is to select the $x_{i,j}$ closest to $\mathcal{C}_{i,j}$ in s_i, giving priority to the one before $\mathcal{C}_{i,j}$. Thus we modify the dependency tree based on the citation nodes.

Algorithm 1 ALiiCE's Parsing Algorithm

Input: A sentence s with inline citation marks
Output: A list of claim of each group of citation marks

$L = \phi$
$s' = \text{TEXTCLEANING}(s)$
$T = \text{DEPENDENCYTREE}(s')$
$nodes = \text{MATCHCITATIONNODES}(T, s)$
for each $node_i$ in $nodes$ **do**
  $T' = \text{DEEPCOPY}(T)$
  for each $node_j$ in $nodes \setminus \{node_i\}$ **do**
    $node_{lca} = \text{LCA}(T', node_i, node_j)$
    $T_i = \text{SUBTREE}(node_{lca}, node_i)$
    $T_j = \text{SUBTREE}(node_{lca}, node_j)$
    if $node_{lca} = node_i \vee$
    $(node_{lca} \neq node_j \wedge T_i < T_j)$ **then**
      $\text{MASK}(T', T_j)$
    else
      $\text{REPLACE}(node_{lca}, T_i)$
  $r = \text{CONVERTTOTEXT}(T')$
  $r \longrightarrow L$
return L

For each citation node, denoted as node i, iterate other citation nodes except node i. When iterating to node j, we calculate the LCA node of node i and node j in T. Then we find the subtrees of LCA node's children containing node i and node j, and denote them as T_i and T_j, respectively. Next, we discuss in different situations:

- If LCA node is node i, remove T_j from T.
- If LCA node is node j, replace LCA node's subtree with T_i.
- If LCA node is another node in T, then we compare the relative positions between T_i and T_j, according to the word's order in the sentence of subtree's root: If T_i is before T_j, then remove T_j from T; If T_i is after T_j, then replace LCA node's subtree with T_i.

After iteration, we obtain a modified dependency tree. We convert words in the modified tree to text following the order in original sentence, getting the claim of citations corresponding to node i. We provide additional details and examples of our algorithm in Appendix A and B, respectively.

3.4 Metrics For Fine-grained Citation

In this section, we display three new metrics for positional fine-grained citation quality in ALiiCE.

3.4.1 Positional Fine-grained Citation Recall

For each $\mathcal{C}_{i,j}$ and its corresponding $\mathcal{A}_{i,j}$, if the concatenation of passages in $\mathcal{C}_{i,j}$ can entail $\mathcal{A}_{i,j}$, then the citation recall of $\mathcal{C}_{i,j}$ is 1, otherwise it is 0. The judgement of entailment can be formulated as:

$$\Psi(\mathcal{H}, \mathcal{S}) = \begin{cases} 1, & \text{if } \mathcal{H} \text{ entails } \mathcal{S} \\ 0, & \text{else} \end{cases} \quad (2)$$

where Ψ represents a NLI model, and $\mathcal{H}$ and $\mathcal{S}$ represent hypothesis and statement, respectively.

3.4.2 Positional Fine-grained Citation Precision

Following (Gao et al., 2023b), we calculate citation precision to evaluate whether every citation is necessary. This metric checks for redundant citations to improve readability and verifiability.

We only calculate citation precision when the citation recall of $\mathcal{C}_{i,j}$ is 1. Specifically, for each $c_{i,j,k}$ in $\mathcal{C}_{i,j}$, if $c_{i,j,k}$ can not entail $\mathcal{A}_{i,j}$ alone while the concatenation of passages in $\mathcal{C}_{i,j} \setminus c_{i,j,k}$ can, it is indicated that $c_{i,j,k}$ is a redundant citation and the precision score of $c_{i,j,k}$ is 0, otherwise the precision score of $c_{i,j,k}$ is 1. Then we calculate the mean of the precision scores from each $c_{i,j,k}$ as the precision score of $\mathcal{C}_{i,j}$. If the citation recall of $\mathcal{C}_{i,j}$ is 0, then its citation precision is 0, as well.

3.4.3 Coefficient of Variation of Citation Positions

Positional fine-grained citation generation allows citation marks to appear anywhere within a sentence, as long as their placement is logical and coherent. This means that the capability of generator $\mathcal{M}$ for positional fine-grained citation can, to some extent, be reflected from the degree of dispersion of citation marks positions. To quantify this capability, we propose CPCV (**C**oefficient of **V**ariation of **C**itation **P**ositions).

For response $\mathcal{R}$, we first calculate the citation position for every sentence. For sentence s_i, the position of citation marks can be expressed as:

$$p_{i,j'} = \frac{j \cdot \mathbb{1}_{s_{i,j} = \mathcal{C}_{i,j}}}{|s_i|} \quad (3)$$

where j' is a new subscript because we omit words from s_i in p_i. And $p_{i,j'}$ ranges from 0 to 1. Then we can calculate the standard deviation of s_i as:

$$\sigma(s_i) = \sqrt{\frac{1}{|p_i|} \sum_{j'=1}^{|p_i|} \left(p_{i,j'} - \mu_i\right)^2} \quad (4)$$

Model (k-psg-form)	ALiiCE		ALCE		Fluency	Correct.	CPCV	Length
	Rec.	Prec.	Rec.	Prec.				
GPT-3.5 (5-psg)	78.4 (0.5)	74.4 (0.4)	80.0 (0.3)	72.9 (0.4)	86.1 (2.9)	51.1 (0.3)	0.10 (–)	50.5 (37.3)
GPT-3.5 (5-psg-summ)	76.9 (0.4)	71.6 (0.9)	77.3 (0.3)	71.5 (0.6)	75.4 (2.3)	49.3 (0.3)	0.13 (–)	40.2 (33.2)
GPT-3.5 (5-psg-snip)	74.4 (0.7)	69.4 (0.3)	74.8 (0.2)	68.2 (0.4)	73.1 (3.7)	48.0 (0.6)	0.13 (–)	36.0 (29.7)
GPT-3.5 (10-psg)	77.7 (1.2)	75.9 (0.8)	79.8 (1.5)	73.8 (1.4)	84.6 (7.1)	44.1 (0.3)	0.15 (–)	63.4 (52.6)
GPT-4 (5-psg)	76.8 (1.2)	68.2 (1.1)	78.5 (0.6)	67.0 (0.6)	52.2 (9.5)	47.0 (0.4)	0.15 (–)	28.1 (20.8)
LLaMA-3-8B (5-psg)	64.8 (1.0)	61.4 (1.4)	65.8 (1.9)	60.3 (1.2)	84.2 (5.0)	50.9 (0.3)	0.44 (–)	64.0 (53.1)
LLaMA-3-8B (10-psg)	61.8 (1.3)	62.5 (0.5)	62.0 (1.3)	60.1 (1.5)	88.8 (9.6)	41.7 (1.4)	0.45 (–)	73.2 (64.9)

Table 1: Results on ASQA. The k-psg indicates using top-k relevant documents for response generation. Document formats include summary (summ), snippet (snip), and default original text. The correctness here refers to the exact match recall. The value in bracket represents the population standard deviation.

Model (k-form)	ALiiCE		ALCE	
	Rec.	Prec.	Rec.	Prec.
GPT-3.5 (5)	75.4 (0.6)	74.2 (0.8)	**80.4** (0.3)	**67.2** (0.8)
GPT-3.5 (5-summ)	73.9 (0.6)	72.4 (0.3)	76.9 (0.3)	59.4 (0.4)
GPT-3.5 (5-snip)	60.5 (0.3)	62.6 (1.0)	68.1 (1.4)	59.4 (1.1)
GPT-3.5 (10)	**75.8** (0.6)	**77.9** (1.0)	78.6 (0.8)	65.6 (0.9)
GPT-4 (5)	69.3 (0.8)	75.7 (0.8)	76.0 (0.3)	66.1 (0.7)
LLaMA-3 (5)	56.9 (1.0)	64.3 (0.4)	60.3 (0.4)	57.9 (1.2)
LLaMA-3 (10)	57.7 (1.0)	66.1 (1.2)	58.2 (1.5)	55.2 (1.4)

Table 2: Results on ASQA when only outputs with positional fine-grained citations are evaluated. We omit the string "-psg" in the model settings for clarity. The best performances are highlighted in bold.

where $\mu_i = \frac{1}{|p_i|}\sum_{j'=1}^{|p_i|} p_{i,j'}$, which represents the average value of p_i. Finally, we give the coefficient of variation of citation positions of response $\mathcal{R}$:

$$CV_{CP}(\mathcal{R}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\sigma(s_i)}{\mu_i} \quad (5)$$

Obviously, a higher CPCV indicates a greater dispersion of citation positions. In practice, there may be sentences in $\mathcal{R}$ that do not have citation markers, thus we do not include these in practice.

4 Experimental Setup

In this section, we describe the datasets and implementation details of our experiments.

4.1 Datasets

We utilize two popular datasets for the task of long-form QA, which are as follows:

ASQA This is an open-domain long-form QA dataset for ambiguous factoid queries, collected from AmbigQA (Min et al., 2020). Each query is annotated with long-form answers and multiple sub query-answer pairs that should be answerable by the long-form answers. We only use the development split of ASQA, which has 948 queries.

ELI5 This is a dataset for long-form QA, collected from subreddit "Explain Like I'm Five". First, its queries are complex enough to encourage paragraph-length responses. Second, each query requires reference to multiple knowledge sources. We only employ 1,000 examples collected randomly from its validation split.

The queries of these two datasets are well suited for retrieval-augmented generation, thus more conducive for evaluating fine-grained citation generation. Following (Gao et al., 2023b), we use the Generalizable T5-based dense Retriever (GTR; Ni et al., 2022) to retrieve relevant passages for queries from Wikipedia corpus snapshot dated 2018-12-20.

4.2 Implementation

In our implementation, we utilize the SpaCy[1] to construct dependency trees for sentences, which is a useful and efficient python toolkit for many NLP tasks. We use TRUE[2], a fine-tuned T5-11B (Raffel et al., 2020) model as the NLI model for the judgement of entailment in citation quality.

4.3 Models

We evaluate both closed-source and open-source LLMs on the task of positional fine-grained citation generation. For closed-source LLMs, we evaluate `gpt-4-turbo-2024-04-09` and `gpt-3.5-turbo-0125` (OpenAI, 2022; OpenAI et al., 2024). For open-source LLMs, we evaluate `LLaMA-3-8B` (AI@Meta, 2024) and set top_p=0.95 for Nucleus Sampling (Holtzman et al., 2020). We set the sampling temperature to 0.5 for all models. In addition,

[1] https://spacy.io/

[2] https://huggingface.co/google/t5_xxl_true_nli_mixture

Model (k-psg-form)	ALiiCE		ALCE		Fluency	Correct.	CPCV	Length
	Rec.	Prec.	Rec.	Prec.				
GPT-3.5 (5-psg)	61.0 (0.5)	58.6 (2.2)	61.9 (0.9)	57.7 (0.6)	21.8 (0.6)	20.8 (0.3)	0.10 (–)	131.7 (46.2)
GPT-3.5 (5-psg-sum)	53.9 (2.6)	52.0 (1.1)	54.7 (2.3)	51.4 (2.5)	21.3 (5.1)	20.8 (1.1)	0.15 (–)	111.3 (46.5)
GPT-3.5 (5-psg-snippet)	53.4 (1.3)	50.9 (1.1)	53.6 (1.1)	50.8 (1.6)	34.9 (7.3)	20.8 (0.4)	0.13 (–)	106.7 (47.9)
GPT-3.5 (10-psg)	58.1 (2.4)	56.8 (2.0)	59.5 (1.7)	55.5 (1.0)	18.5 (4.7)	19.7 (0.7)	0.12 (–)	155.9 (57.4)
GPT-4 (5-psg)	55.1 (0.5)	54.0 (3.0)	57.9 (2.2)	53.2 (2.1)	20.4 (7.2)	21.3 (0.9)	0.15 (–)	102.2 (59.7)
LLaMA-3-8B (5-psg)	45.9 (0.3)	47.1 (0.7)	47.9 (2.1)	46.3 (1.0)	36.2 (1.0)	20.5 (0.9)	0.53 (–)	203.9 (71.4)
LLaMA-3-8B (10-psg)	42.8 (0.8)	44.2 (0.9)	43.6 (1.7)	42.4 (0.5)	32.5 (6.2)	19.5 (0.7)	0.61 (–)	224.2 (77.7)

Table 3: Results on ELI5. The correctness here refers to the exact match recall. Other descriptions follow Table 1.

Model (k-psg-form)	ALiiCE		ALCE	
	Rec.	Prec.	Rec.	Prec.
GPT-3.5 (5)	40.1 (1.8)	**50.0** (2.8)	**48.1** (2.4)	**44.2** (2.2)
GPT-3.5 (5-summ)	35.9 (2.5)	35.1 (2.6)	42.9 (1.2)	32.4 (1.5)
GPT-3.5 (5-snip)	39.6 (2.4)	39.1 (1.3)	43.4 (3.5)	34.5 (2.7)
GPT-3.5 (10)	**44.2** (0.7)	48.1 (1.0)	46.7 (0.5)	41.0 (1.7)
GPT-4 (5)	40.5 (1.9)	46.2 (1.1)	44.6 (3.6)	38.7 (3.5)
LLaMA-3 (5)	41.1 (1.6)	44.0 (0.9)	43.4 (1.5)	39.7 (1.0)
LLaMA-3 (10)	41.8 (1.7)	47.7 (3.7)	43.6 (3.0)	41.0 (3.6)

Table 4: Results on ELI5 when only outputs with positional fine-grained citations are evaluated. Other descriptions follow Table 2.

we also incorporate variables such as the number of retrieved documents and the document form used in generation (truncated original text, summary, or snippet) into the model setting. All prompts we employed are provided in Appendix C.

4.4 Evaluation Metrics

For comparison, we evaluate ALiiCE, representing the positional fine-grained citation evaluation method, and ALCE (Gao et al., 2023b), representing the sentence-level citation evaluation method.

In addition to the citation-specific metrics introduced above, we also utilize some common metrics in long-form QA. These metrics can also reflect potential influences caused by positional fine-grained citations, as detailed below:

Correctness We check whether $\mathcal{R}$ answers the query q accurately. For ASQA, we following (Stelmakh et al., 2022) to calculate exact match recall by checking whether ground truths are exact substrings of $\mathcal{R}$; For ELI5, we following (Fan et al., 2019) to use the F1 score of ROUGE-L.

Fluency We quantify the fluency of text by MAUVE (Pillutla et al., 2021) to evaluate whether $\mathcal{R}$ generated by $\mathcal{M}$ is coherent, which is also essential for the task of long-form QA.

Length We calculate the average length of $\mathcal{R}$ since it is sensitive to the long-form answers.

5 Main Results

The main results of our experiments on the ASQA and ELI5 datasets are presented in Table 1 and Table 3, respectively. In this section, we firstly present our key observations on the results, and then provide our detailed analyses on cases to show the shortcomings of sentence-level evaluation and the effectiveness of our approach.

5.1 Overall Performances

Difference in metric judgement. There are differences in the evaluation on positional fine-grained citation text generation between ALiiCE and ALCE. Compared to ALCE, ALiiCE calculates lower citation recall, but higher citation precision. And this difference becomes more dramatic when only positional fine-grained citation outputs are evaluated, as illustrated in Table 2 and Table 4. We can observe this more intuitively in Figure 3. This change also leads to different best performing models under the two evaluation methods, according to the citation quality. GPT-3.5 with 5-passages setting performs best under the evaluation of ALCE, while ALiiCE tends to favor GPT-3.5 with 10-passages setting.

Difficulty in fine-grained generation. We find that existing LLMs face challenges in generating positional fine-grained citations. LLMs still generate fewer fine-grained samples when prompts contain related documents. Although we employ In-Context Learning (ICL; Dong et al., 2023) to guide LLMs in placing citation marks by demonstration, this only produces slight improvement. The lower CPCV is indicative of this, which explains why ALCE and ALiiCE do not show very significant differences when evaluated on all citation texts. We suggest that positional fine-grained citation generation might be more suitable for specialized domains rather than open-domain QA.

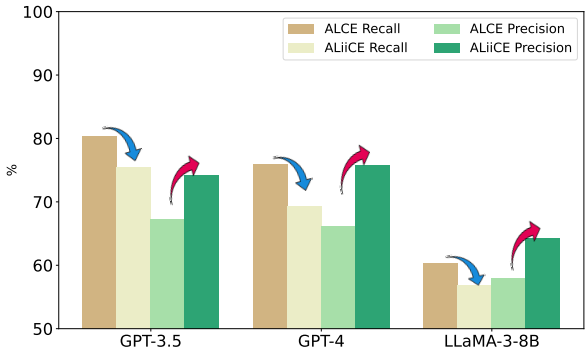

Figure 3: Comparison of citation recall and precision between ALCE and ALiiCE across three models on the ASQA dataset. All models use the 5-psg setting. The arrows indicate a uniform law: ALiiCE achieves a lower citation recall and higher citation precision.

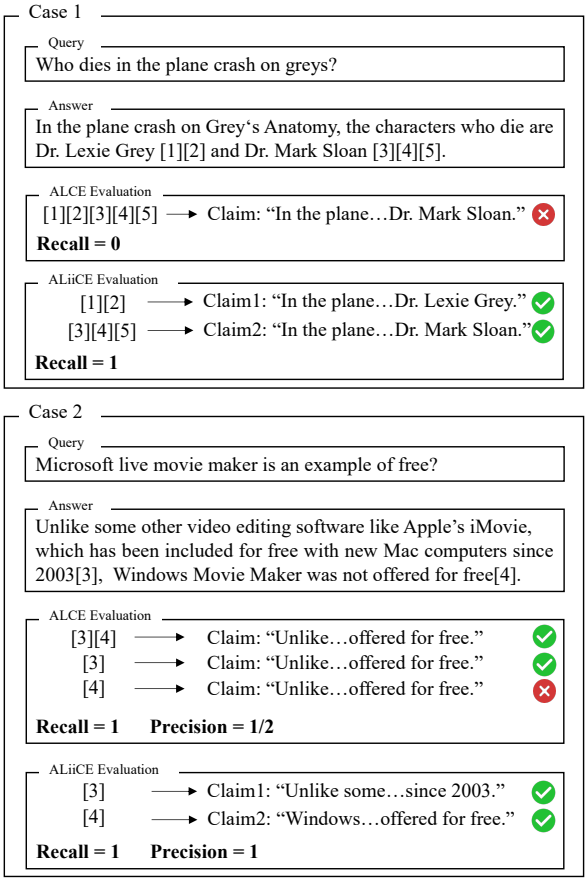

Figure 4: Evaluation of citation quality by ALCE and ALiiCE on two examples from ASQA. The answers are generated by GPT-3.5 (5-psg). Case 1 demonstrates the difference between two methods on citation recall, and Case 2 illustrates the difference on citation precision.

Advances in open-source LLMs. LLaMA-3 narrows the gap between open-source LLMs and closed-source LLMs in the citation text generation task. In previous studies, the citation quality of open-source LLMs is significantly worse than that of closed-source LLMs (Gao et al., 2023b; Huang et al., 2024). However, our experimental results show that the citation recall and precision of GPT-4 with 5-passages are only improved by 20.0% and 14.6%, respectively, compared to LLaMA-3-8B with 5-passages on ELI5. Additionally, LLaMA-3-8B has a higher CPCV and exhibits greater fluency, than both GPT-3.5 and GPT-4.

5.2 Case Study

Long-context issue. Sentence-level evaluation can result in inaccuracies when dealing with long-context NLI. For instance, in Case 1 depicted in Figure 4, when assessing citation recall, the concatenated passages exceed the context length of NLI model, potentially leading to incorrect inference results due to distracted attention or truncation of evidences. In ALiiCE, evidences are dispersed by parsing atomic claims, reducing the likelihood of exceeding context limits.

Citation precision issue. There is a shortcoming in the calculation of citation precision in sentence-level evaluation, when assessing positional fine-grained citations. If there is an overlap between different evidences, it is potential for the NLI model to misjudge multiple atomic claims simultaneously. Taking the Case 2 in Figure 4 as an example, citation "[3]" contains evidences supporting both atomic claim 1 and 2. According to ALCE's citation precision, citation "[3]" alone can support the entire sentence-level claim, whereas citation "[4]" cannot, as it only supports atomic claim 2. Consequently, citation "[4]" is considered redundant, despite it is actually a reasonable citation. In ALiiCE, we evaluate based on atomic claims, ensuring that the assessment is not influenced by evidences from other claims.

Based on the observed phenomenon of lower recall and higher precision, along with the previous analyses, we conclude that: compared to other metrics, ALiiCE has a higher decision threshold in the citation generation task. This indicates that ALiiCE is more conservative, only considering a citation correct when it has a high level of confidence. This is more beneficial for the citation generation task because the higher decision threshold encourages more accurate and relevant citations, reducing the likelihood of misleading information, which is particularly crucial in professional and high-risk fields (such as law and medicine) where incorrect cita-

Dataset	Num of Claims	Num of same NLI
ASQA	1935	1930
ELI5	3923	3891

Table 5: Results on parsing error analyses. The second column is the total number of claims. The last column is the number of claims with consistent NLI results before and after refinement on the claims.

tions can lead to serious consequences. Therefore, ALiiCE can help further enhance LLMs' credibility while enabling effective evaluation.

5.3 Error Analyses

We further analyzed the potential errors in ALiiCE, which mainly come from two aspects:

Grammatical error. Grammatical errors in the sentence can lead to inaccurate parsing results. However, current LLMs exhibit strong grammatical capabilities (Zhao et al., 2023), and after our manual evaluation, the number of samples containing grammatical errors in LLMs' outputs is nearly zero, thus this type of error can be ignored.

Parsing error. Dependency tree parsing itself might contain errors. For example, in sentence "Other radiological signs of fetal death include gas in the fetus or in the portal and umbilical vessels [1], and Deuel's halo sign [2].", the atomic claim of citation "[2]" is parsed as "Other radiological signs of fetal death include gas Deuel 's halo sign" by SpaCy, which contains an extra word "gas" due to an error from dependency recognition.

Therefore, we conduct further experiment to test the potential impact of parsing errors on NLI. We firstly collect all the atomic claims from two datasets. Next, we utilize GPT-3.5 to refine each claim based on its original sentence (the prompt is provided in Appendix C). And then we employ the NLI model to assess the entailment before and after the claim refinement. As indicated in Table 5, the result show that the proportion of claims with inconsistent NLI results is less than 1% across both datasets. Therefore, the parsing error is unlikely to have a significant impact on the evaluation.

6 Related Work

Attribution. Attribution refers to the ability of LMs to generate and provide evidence (Li et al., 2023). The source of attribution can be pre-training data (Han and Tsvetkov, 2022; Weller et al., 2024), or out-of-model knowledge (Shuster et al., 2021; Li et al., 2024). When the source is documents, citation is a common form of attribution (Kamalloo et al., 2023). Ye et al. (2024); Huang et al. (2024) study generating response and citations simultaneously, while Gao et al. (2023a); Huo et al. (2023) research on adding citations in the post-hoc stage.

Retrieval-Augmented Generation. Retrieval-augmented generation (RAG; Lewis et al., 2020) combines the strengths of information retrieval and generation models, demonstrating improvement in several NLP tasks. The primary methods for incorporating external knowledge into generation include modifying model parameters (Sen et al., 2023) and Chain-of-Thought (CoT; Wei et al., 2022; Xu et al., 2024). Since RAG exhibits a black-box nature (Gao et al., 2024), adding citations in response can effectively mitigate the hallucination problem and enhance verifiability.

Citation Evaluation. The current citation evaluation methods are mainly performed by human evaluation, which is costly and time-intensive (Chen et al., 2023). Subsequently, automatic evaluation methods are proposed, including classification-based metrics (Liu et al., 2023; Yue et al., 2023) and quantitative metrics (Gao et al., 2023b; Li et al., 2024). However, these methods are primarily sentence-level, leading to issues with atomicity of claims (Malaviya et al., 2024) and ambiguity (Liu et al., 2023). We propose ALiiCE, the first evaluation method for positional fine-grained citations.

7 Conclusion

In this study, we propose ALiiCE, the first evaluation method for positional fine-grained citation generation. Our approach incorporates an algorithm for parsing atomic claims based on dependency analysis, along with three metrics designed to assess the quality of positional fine-grained citations.

We evaluate different LLMs and obtain several observations: 1) ALiiCE shows a higher decision threshold compared to sentence-level evaluation; 2) existing LLMs face challenges in generating positional fine-grained citations; 3) open-source LLMs narrow the gap in citation generation with closed-source LLMs. Our further analyses demonstrate insufficiency of existing sentence-level evaluation methods and effectiveness of our ALiiCE for assessing positional fine-grained citations. We hope that our work can inspire more research into positional fine-grained citation text generation.

Limitations

In the implementation of our parsing method, we only employ SpaCy to construct dependency trees. Other dependency analysis methods with higher accuracy can improve our benchmark, which are not evaluated in our work. In addition, dependency analysis may be primarily applicable to mainstream languages such as English. Thus directly transferring ALiiCE to other languages might result in reduced evaluation accuracy.

In our experiments, we only utilize the open-domain long-form QA datasets. However, positional fine-grained citation generation is applicable to a broader range of scenarios, such as academic writing and automatic summarization. Additionally, positional fine-grained citations are more likely to emerge when there are clear logical relationships between claims, such as multi-hop reasoning. Therefore, it is necessary to expand the data domain of the benchmark.

Ethics Statement

The citation generation task aims to enhance the credibility of the generative model, assist users in verifying information, and mitigate the spread of misunderstandings or incorrect information. Additionally, it helps reduce ethical risks by clarifying responsibilities and respecting intellectual property rights. This research utilizes publicly available datasets sourced from widely recognized and reputable repositories. We have ensured that all datasets used in this study comply with relevant data usage and privacy policies.

References

AI@Meta. 2024. Llama 3 model card.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.

Hung-Ting Chen, Fangyuan Xu, Shane Arora, and Eunsol Choi. 2023. Understanding retrieval augmentation for long-form question answering. Preprint, arXiv:2310.12150.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pages 423–429, Barcelona, Spain.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning. Preprint, arXiv:2301.00234.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Martin Funkquist, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. 2023. Citebench: A benchmark for scientific citation text generation. Preprint, arXiv:2212.09577.

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.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. Preprint, arXiv:2312.10997.

Xiaochuang Han and Yulia Tsvetkov. 2022. Orca: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data. Preprint, arXiv:2205.12600.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. Preprint, arXiv:1904.09751.

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.

Chengyu Huang, Zeqiu Wu, Yushi Hu, and Wenya Wang. 2024. Training language models to generate text with citations via fine-grained rewards. Preprint, arXiv:2402.04315.

Siqing Huo, Negar Arabzadeh, and Charles Clarke. 2023. Retrieving supporting evidence for generative question answering. In Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP '23. ACM.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. Preprint, arXiv:2307.16883.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4940–4957, Online. 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, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Advances in Neural Information Processing Systems, volume 33, pages 9459–9474. Curran Associates, Inc.

Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. 2023. A survey of large language models attribution. Preprint, arXiv:2311.03731.

Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. 2024. Towards verifiable generation: A benchmark for knowledge-aware language model attribution. Preprint, arXiv:2310.05634.

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.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. Expertqa: Expert-curated questions and attributed answers. Preprint, arXiv:2309.07852.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5783–5797, Online. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt blog post.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2024. Gpt-4 technical report. Preprint, arXiv:2303.08774.

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 Advances in Neural Information Processing Systems, volume 34, pages 4816–4828. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67.

Priyanka Sen, Sandeep Mavadia, and Amir Saffari. 2023. Knowledge graph-augmented language models for complex question answering. In Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE), pages 1–8, Toronto, Canada. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. 2024. Attribute first, then generate: Locally-attributable grounded text generation. Preprint, arXiv:2403.17104.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems, volume 35, pages 24824–24837. Curran Associates, Inc.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. "according to . . . ": Prompting language models improves quoting from pre-training data. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2288–2301, St. Julian's, Malta. Association for Computational Linguistics.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. Preprint, arXiv:2304.14732.

Xi Ye, Ruoxi Sun, Sercan Ö. Arik, and Tomas Pfister. 2024. Effective large language model adaptation for improved grounding and citation generation. Preprint, arXiv:2311.09533.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 4615–4635, Singapore. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. Preprint, arXiv:2303.18223.

A Parsing Algorithm Details

In section 3.3, we simply the process of parsing algorithm. In practice, we consider more details when decomposing modified trees for different claims. The dependency type, represented by the edge values in the dependency tree (which can refer to Figure 5), is a crucial factor in dependency analysis. Thus we take dependency types into account when modifying the dependency tree. Table 6 shows some common dependency types, and a comprehensive explanation can be found in the official SpaCy documentation[3].

Specifically, when calculating the modified tree for node i and traversing to node j in iteration, if the LCA node is neither node i nor node j, a more detailed discussion by situations is as follows:

- If there is a subtree between T_i and T_j with a dependency relation of "cc" between its root node and the LCA node (we refer to this subtree T_c), then we discuss
  - If T_i is before T_j, then we discuss: If the LCA node is the root node of the dependency tree and T_i has a dependency relation of "prep" or "advcl" with the LCA node, then replace the root node of the dependency tree with T_i; else, then remove T_j and T_c.
  - If T_i is after T_j, then we discuss: If the LCA node is the root node of the dependency tree and T_i has a dependency relation of "prep" or "advcl" with the LCA node, then remove T_j and T_c; else, then replace the root node of the dependency tree with T_i.
- Else, then we discuss: If the LCA node is the root node of the dependency tree, then replace the root node of the dependency tree with T_i; else, then remove T_j from T.

B Parsing Examples

To improve the intuitiveness of the parsing algorithm, we present three straightforward examples (Figures 5 to 11). Each figure shows a dependency tree, where each node represents a word node. For word nodes matched with citations (marked in red), the format of the node value is "word : index : citation marks", where "index" denotes the position of the word in the original sentence. For word nodes without citations (marked in green), the format of the node value is "word : index". The sentences to be parsed are all from the outputs of the GPT-3.5 (5-psg) on the ASQA dataset.

Specifically, Figure 5 illustrates the dependency tree for "In the plane crash on Grey's Anatomy, the characters who die are Dr. Lexie Grey [1][2] and Dr. Mark Sloan [3][4][5].", and Figures 6 and 7 display the modified trees for the two atomic claims in the output. Similarly, Figure 8, 9, and 10 correspond to output "Some brands, such as Export As, come in packs of 25 [2], while standard packs typically contain 20 cigarettes [4].", and Figure 11, 12, and 13 correspond to output "Queen Victoria became Queen of the United Kingdom on 20 June 1837[3], while Queen Anne became Queen of England, Scotland, and Ireland on 8 March 1702[1].". Notably, in the dependency tree shown in Figure 5, the LCA node of the two citation nodes is one of them. This structure represents the parallel relationship between two claims, which is a common form in positional fine-grained citations.

Relation Type	Explanation
acomp	adjectival complement
advcl	adverbial clause modifier
amod	adjectival modifier
cc	coordination
conj	conjunct
mark	marker
nmod	nominal modifier
nsubj	nominal subject
nsubjpass	passive nominal subject
pobj	object of a preposition
prep	prepositional modifier
punct	punctuation
relcl	adnominal relative clause modifier

Table 6: Several common types of dependency relation.

C Prompts

We provide the prompts used in our experiments. We utilize the same prompt in fine-grained citation generation for all models, as shown in Table 7. And Table 8 shows the prompt for claim rewriting employed in our error analysis experiments.

[3] https://spacy.io/api/dependencyparser

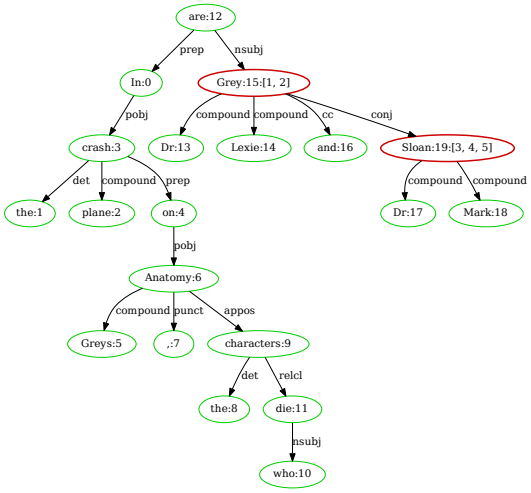

Figure 5: The dependency tree of sentence "In the plane crash on Grey's Anatomy, the characters who die are Dr. Lexie Grey [1][2] and Dr. Mark Sloan [3][4][5].", from the response generated by GPT-3.5 (5-psg). The query is "Who dies in the plane crash on greys?" from ASQA. The modified tree of claim corresponds to citation "[1][2]" is shown at Figure 6. The modified tree of claim corresponds to citation "[3][4][5]" is shown at Figure 7.

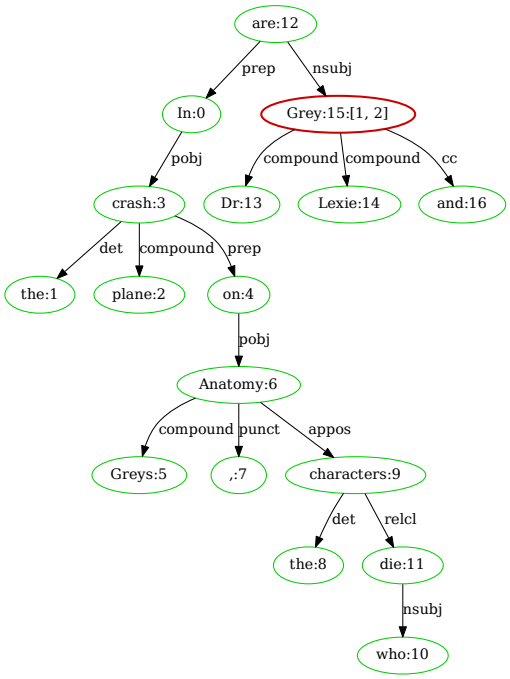

Figure 6: The modified tree of claim "In the plane crash on Greys Anatomy , the characters who die are Dr Lexie Grey and". This claim corresponds to citation "[1][2]" of sentence which is illustrated in Figure 5.

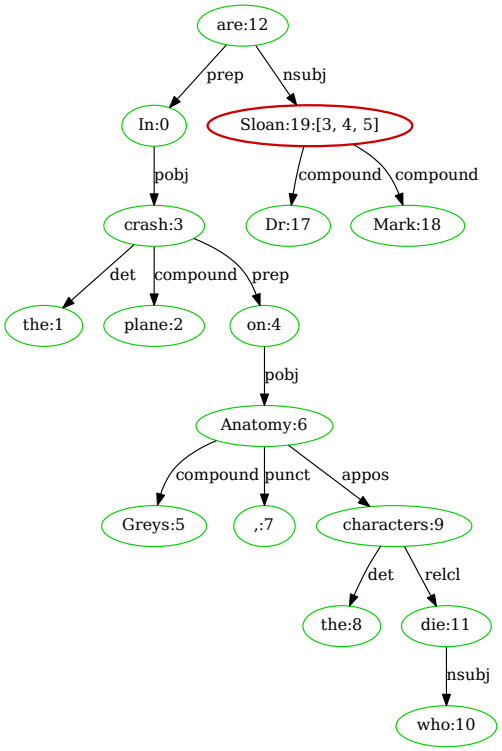

Figure 7: The modified tree of claim "In the plane crash on Greys Anatomy , the characters who die are Dr Mark Sloan". This claim corresponds to citation "[3][4][5]" of sentence which is illustrated in Figure 5.

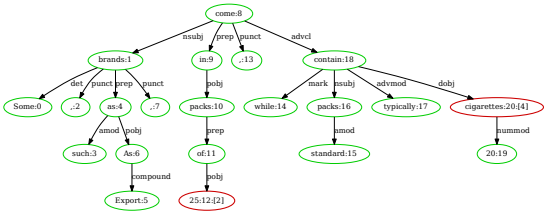

Figure 8: The dependency tree of sentence "Some brands, such as Export As, come in packs of 25 [2], while standard packs typically contain 20 cigarettes [4].", from the response generated by GPT-3.5 (5-psg). The query is "Number of cigarettes in a pack in usa?" from ASQA. The modified tree of claim corresponds to citation "[2]" is shown at Figure 9. The modified tree of claim corresponds to citation "[4]" is shown at Figure 10.

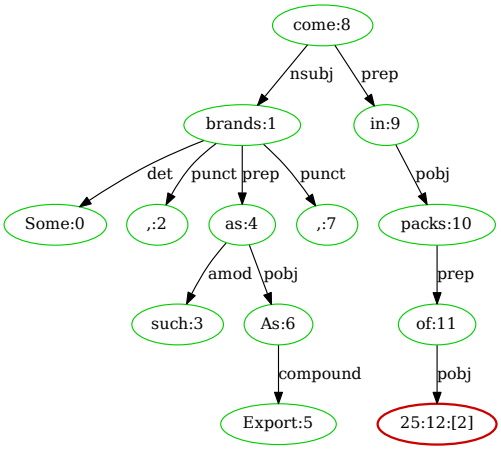

Figure 9: The modified tree of claim "Some brands , such as Export As , come in packs of 25". This claim corresponds to citation "[2]" of sentence which is illustrated in Figure 8.

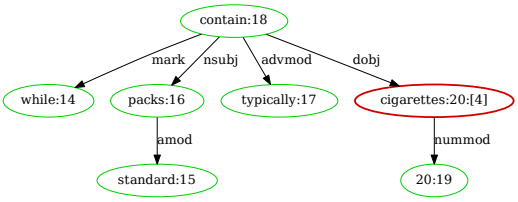

Figure 10: The modified tree of claim "while standard packs typically contain 20 cigarettes". This claim corresponds to citation "[4]" of sentence which is illustrated in Figure 8.

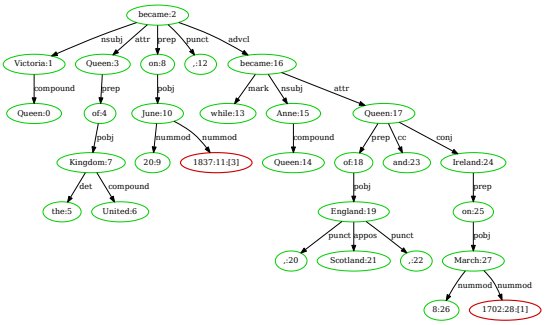

Figure 11: The dependency tree of sentence "Queen Victoria became Queen of the United Kingdom on 20 June 1837[3], while Queen Anne became Queen of England, Scotland, and Ireland on 8 March 1702[1].", from the response generated by GPT-3.5 (5-psg). The query is "When did the queen became queen of england?" from ASQA. The modified tree of claim corresponds to citation "[3]" is shown at Figure 12. The modified tree of claim corresponds to citation "[1]" is shown at Figure 13.

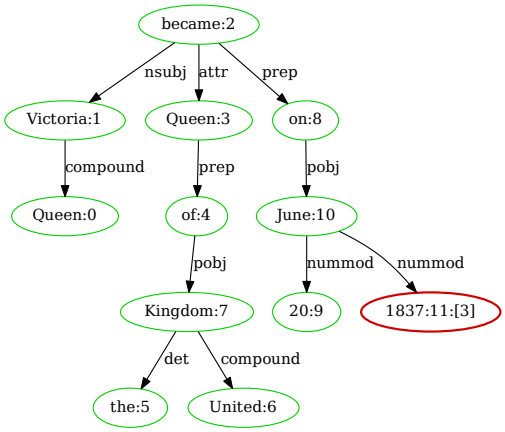

Figure 12: The modified tree of claim "Queen Victoria became Queen of the United Kingdom on 20 June 1837". This claim corresponds to citation "[3]" of sentence which is illustrated in Figure 11.

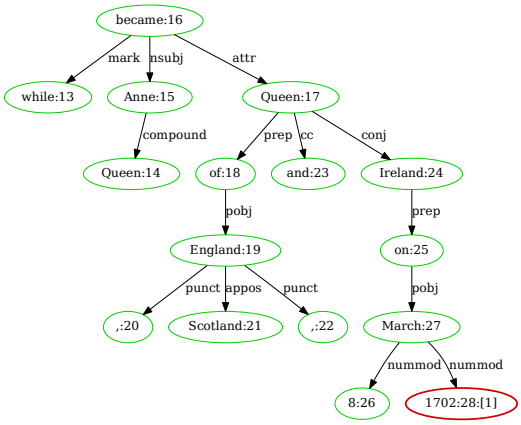

Figure 13: The modified tree of claim "while Queen Anne became Queen of England , Scotland , and Ireland on 8 March 1702". This claim corresponds to citation "[1]" of sentence which is illustrated in Figure 11.

Instruction: Please provide an accurate and concise answer that includes fine-grained in-text citations immediately following the relevant information. Place the citation numbers within brackets directly after the facts they support.

Citation format examples:
1. One of the most important areas is the automatic detection of vandalism[1][3] and data quality assessment in Wikipedia[2][4].
2. Cups can be made of glass[1] or plastic[2][3].Wikipedia's community has been described as cultlike[1], although not always with entirely negative connotations[2].
3. Wikipedia's community has been described as cultlike[1], although not always with entirely negative connotations[2].

Question: Who gets fired on grey's anatomy season 6?

Documents [1] (Title: Now or Never (Grey's Anatomy)) an accident during the episode and dies in the season 6 premier. In the episode Cristina Yang (Sandra Oh), Alex Karev (Justin Chambers), George O'Malley (T.R. Knight), and Meredith Grey (Ellen Pompeo) are all sleeping and waiting for Izzie Stevens (Katherine Heigl) to wake up after the surgery. Derek Shepherd (Patrick Dempsey) comes up with an alternative treatment plan for Izzie, Miranda Bailey (Chandra Wilson) confronts Chief's Richard Webber (James Pickens Jr.) and Arizona Robbins (Jessica Capshaw), about the peds fellowship program. Yang deals with her relationship with Owen Hunt (Kevin McKidd) who helps George with career advice. The episode

Documents [2] (Title: Grey's Anatomy) the head of neurosurgery and Meredith's love interest; Preston Burke (Isaiah Washington), the head of cardio, who becomes Yang's fiancé; and Richard Webber (James Pickens, Jr.), the Chief of Surgery and attending general surgeon, and the previous lover of Ellis Grey. In the sixth season, these residents are joined by Jackson Avery (Jesse Williams) and April Kepner (Sarah Drew), former Mercy-West residents who join Seattle Grace following an administrative merger. During the first six seasons, Burke, O'Malley, and Stevens all depart the series. In addition to Webber, Burke, and Shepherd, the surgical wing is primarily supervised by Addison Montgomery (Kate

Documents [3] (Title: Grey's Anatomy (season 6)) Grey's Anatomy (season 6) The sixth season of the American television medical drama "Grey's Anatomy," commenced airing on the American Broadcasting Company (ABC) in the United States on September 24, 2009, and concluded on May 20, 2010. The season was produced by ABC Studios, in association with Shondaland Production Company and The Mark Gordon Company; the showrunner being Shonda Rhimes. Actors Ellen Pompeo, Sandra Oh, Katherine Heigl, and Justin Chambers reprised their roles as surgical residents Meredith Grey, Cristina Yang, Izzie Stevens, and Alex Karev, respectively. Heigl was released from her contract in the middle of the season, while T.R.

...

Answer: In "Grey's Anatomy" Season 6, the characters who get fired include Preston Burke, the head of cardio[2], and Izzie Stevens, portrayed by Katherine Heigl, who was released from her contract in the middle of the season[3]. Additionally, during the first six seasons, Burke, George O'Malley, and Izzie Stevens all depart the series[2].

Table 7: The prompt used to generate a response. The blue text indicates the output of GPT-3.5. The question is from ASQA and the documents is retrieved from Wikipedia corpus by GTR.

Instruction: The following sentence may have some grammatical errors and may have some redundant ingredients. As long as it ensures fluency, you can delete some parts of the sentence that you think don't make sense.

Original sentence: Other radiological signs of fetal death include gas in the fetus or in the portal and umbilical vessels, and Deuel's halo sign.

Sentence to modify: Other radiological signs of fetal death include gas Deuel 's halo sign

Modified sentence: Other radiological signs of fetal death include Deuel's halo sign.

Table 8: The prompt used to refine a claim. The blue text indicates the output of GPT-3.5.