# WebCiteS: Attributed Query-Focused Summarization on Chinese Web Search Results with Citations

Anonymous ACL submission

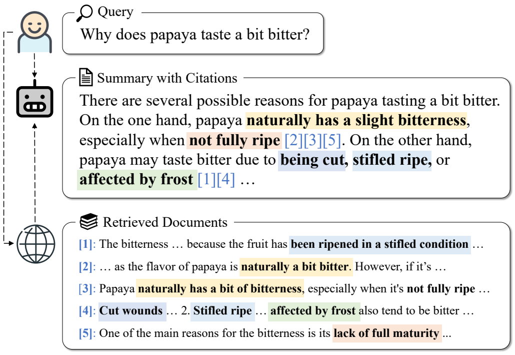
## Abstract

Enhancing the attribution in large language models (LLMs) is a crucial task. One feasible approach is to enable LLMs to cite external sources that support their generations. However, existing datasets and evaluation methods in this domain still exhibit notable limitations. In this work, we formulate the task of attributed query-focused summarization (AQFS) and present WebCiteS, a Chinese dataset featuring 7k human-annotated summaries with citations. WebCiteS derives from real-world user queries and web search results, offering a valuable resource for model training and evaluation. Prior works in attribution evaluation do not differentiate between groundedness errors and citation errors. They also fall short in automatically verifying sentences that draw partial support from multiple sources. We tackle these issues by developing detailed metrics and enabling the automatic evaluator to decompose the sentences into sub-claims for fine-grained verification. Our comprehensive evaluation of both open-source and proprietary models on WebCiteS highlights the challenge LLMs face in correctly citing sources, underscoring the necessity for further improvement. The dataset and code will be open-sourced to facilitate further research in this crucial field.

## 1 Introduction

In today's information-driven society, swift access to knowledge is essential. A major limitation of web search engines is the need for users to manually compile information from various sources, which can be time-consuming. Large language models (LLMs) (Zhao et al., 2023) exhibit potential in this domain by generating straightforward and well-organized responses. However, the potential risks of hallucination (Ji et al., 2023; Zhang et al., 2023c) and factual errors (Min et al., 2023) undermine their trustworthiness as knowledge sources.

An emerging solution is *generative search engines* (Liu et al., 2023a) which use LLMs to synthesize web search results into responses with in-line citations. This allows users and developers to verify the generations against the cited sources. However, recent investigations on commercial products and retrieval-augmented LLMs reveal frequent occurrences of unsupported claims and incorrect citations (Liu et al., 2023a; Gao et al., 2023b), highlighting the challenges of attribution in LLMs (Li et al., 2023a). Nonetheless, the limitations of pertinent datasets and evaluation methods pose obstacles to in-depth explorations within the community.



Why does papaya taste a bit bitter?

There are several possible reasons for papaya tasting a bit bitter. On the one hand, papaya **naturally has a slight bitterness**, especially when **not fully ripe** [2][3][5]. On the other hand, papaya may taste bitter due to **being cut**, **stifled ripe,** or **affected by frost** [1][4] …

[1]: The bitterness … because the fruit has **been ripened in a stifled condition** …

[2]: … as the flavor of papaya is **naturally a bit bitter**. However, if it's …

[3]: Papaya **naturally has a bit of bitterness**, especially when it's **not fully ripe** …

[4]: **Cut wounds** … 2. **Stifled ripe** … **affected by frost** also tend to be bitter …

[5]: One of the main reasons for the bitterness is its **lack of full maturity** ...

Figure 1: Illustration of attributed query-focused summarization (AQFS). Full example is shown in Table 10.

**Firstly, most existing datasets are deficient in high-quality citation annotations.** For instance, the ALCE benchmark (Gao et al., 2023b) compiles three question-answering datasets (Fan et al., 2019; Stelmakh et al., 2022; Amouyal et al., 2023) without providing citations in the reference answers, limiting its utility for model training. In contrast, WebGLM (Liu et al., 2023b) prompts InstructGPT (Ouyang et al., 2022) to generate training data with citations. It controls the citation quality via a sample filtering method which calculates the Rouge (Lin, 2004) score between the answers and their citations. However, this method focuses on lexical similarity rather than logical entailment, and thus could not precisely measure attribution.

**Secondly, current evaluation methods are insufficient to thoroughly assess attribution.** Prior

|  | Lan. | Query Source | Document Source | # Cnt. | Docs Length | Response Length | # Citations per Sent. | Citation Annot. | Citation Eval. |
|---|---|---|---|---|---|---|---|---|---|
| WebCiteS | ZH | Real user queries | Web search results | 7k | 1025 / 3970 | 167 | 1.55 | human | auto |
| WebCPM (Qin et al., 2023) | ZH | Translated Reddit questions | Web searh results | 5k | 513† | 244† | 0.34 | human | N/A |
| WebGLM (Liu et al., 2023b) | EN | ELI5 | Web searh results | 45k | 304 | 104 | 1.20 | auto | human |
| ALCE (Gao et al., 2023b) | EN | ASQA QAMPARI, ELI5 | Wikipedia Common Crawl | 3k | 100*k | 78 | N/A | N/A | auto |

Table 1: Comparison of WebCiteS and relevant datasets. Docs length refers to the total length of the input documents per query. WebCiteS offers two document settings: snippets or full content. The length is measured in characters for Chinese and in words for English. † denotes a number reported in the respective paper; otherwise, it's our calculation using open-source data. ALCE limits document length to 100 and varies the number of retrieved documents from 3 to 20. Moreover, it does not offer golden documents for each query and citation annotations. As for the evaluation of citation quality, WebCPM does not consider citation in its evaluation, WebGLM employs human ratings of citation accuracy, and ALCE offers automatic metrics with limitations we seek to address in this work.

works only inspect if the generations are supported by their citations (Liu et al., 2023a; Gao et al., 2023b; Liu et al., 2023b) without checking all the documents provided in the input context. However, instances of unsupported generations may result from both **failing to correctly cite supporting documents** and **failing to be grounded in all input documents**. Differentiating these two types of errors is crucial for system optimization. Moreover, existing automatic evaluation (Gao et al., 2023b) solely relies on off-the-shelf natural language inference (NLI) models which only recognize entailment (full support) and overlook sentences with multiple sub-claims drawing **partial support** from different sources. Such complexities are common in real-world scenarios (Chen et al., 2023; Kamoi et al., 2023) and are indicative of a strong capability of synthesizing information across various sources.

To address the above limitations, we present **WebCiteS**, a Chinese dataset for **Attributed Query-Focused Summarization (AQFS)**. As shown in Figure 1, given a query and the retrieved documents, AQFS aims to summarize all pertinent information from the documents with in-line citations to make the summary attributable. Our dataset is built upon real-world user queries and search results from a widely used Chinese web search engine. We employ human efforts to ensure the quality of summaries and citations. Table 1 presents a comparison between WebCiteS and relevant datasets.

We propose a comprehensive evaluation framework with a cost-effective automatic evaluator. Our evaluation metrics distinguish two key aspects: **groundedness** (if the model outputs are contextually supported) and **citation quality** (citation accuracy and comprehensiveness), enabling a more nuanced understanding of attribution errors. We also train a tailored claim-split model to extract the sub-claims of a sentence for fine-grained verification. This allows the detection of partial support and improves the alignment between our automatic evaluator and human citations.

Our evaluation of both open-source and proprietary models on WebCiteS reveals the following key findings: (1) contextual grounding of generations does not guarantee the avoidance of citation errors, indicating the challenge of explicit attribution in all the tested LLMs; (2) although supervised fine-tuning improves both groundedness and citation quality, the top-performing model only reaches a citation $F_1$ score of 76.1% with about 20% of sentences not being fully supported by their citations, underscoring the need for further optimization; (3) models perform worse when summarizing full content of web pages rather than shorter snippets, showing that LLMs are less effective at synthesizing and attributing information in the longer context; (4) making documents more fine-grained leads to poorer attribution results, highlighting the difficulty LLMs face in pinpointing the exact supporting evidence within the context.

The dataset and code will be made publicly available for further exploration in this crucial field.

## 2 The WebCiteS Dataset

In this section, we first formulate the AQFS task and then delineate the construction of WebCiteS.

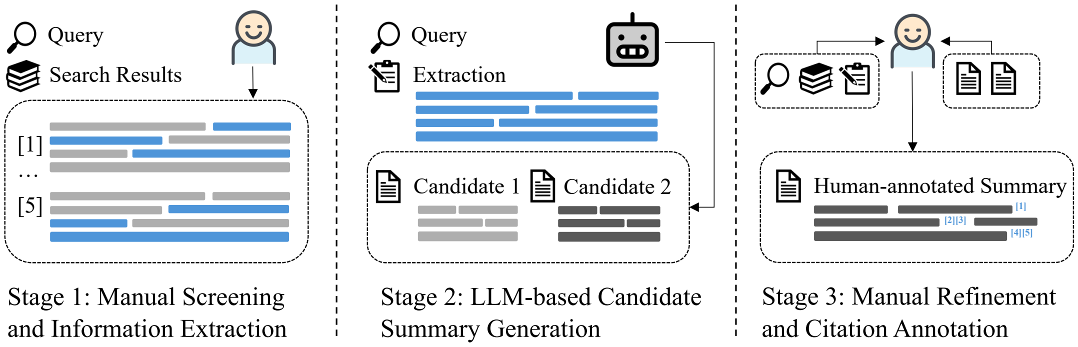

Query
Search Results
[1]
…
[5]
Stage 1: Manual Screening and Information Extraction
Query
Extraction
Candidate 1
Candidate 2
Stage 2: LLM-based Candidate Summary Generation
Human-annotated Summary
[1]
[2][3]
[4][5]
Stage 3: Manual Refinement and Citation Annotation

Figure 2: Illustration of the human-LLM collaborative annotation pipeline of WebCiteS. Initially, annotators manually extract useful information from the documents; then, LLMs are used to generate candidate summaries from the extraction; finally, annotators choose the preferred candidate, refine its quality, and annotate citations.

## 2.1 Task Formulation of AQFS

For a query $q$ and its set of retrieved documents $\mathcal{D}$, AQFS aims to generate a summary $\mathcal{S}$. Following previous works (Liu et al., 2023a; Gao et al., 2023b), we segment it into sentences: $\mathcal{S} = \{s_1, \ldots, s_n\}$. Each sentence $s_i$ cites a subset of documents $\mathcal{C}_i = \{d_1, d_2, \ldots\}$ where $d_i \in \mathcal{D}$. Citations are only required for sentences deemed *verification-worthy* (Liu et al., 2023a), i.e., sentences needing external evidence for validation. We formulate this property with *citation mask* $\mathcal{M} = \{m_1, ..., m_n\}$, where $m_i$ is a binary value and $m_i = 1$ denotes sentence $s_i$ requires citations. In practice, we find most of the sentences require citations, except the introductory or concluding sentences in the summary, such as *"There are several possible reasons for papaya tasting a bit bitter"*.

## 2.2 Data Collection

We collected 40,000 unique and anonymized user queries from the log of a widely used Chinese web search engine. After initial refinement, we retained 18,500 non-trivial queries and retrieved five web pages for each query. The snippets of web pages returned by the search engine were expanded to a maximum of 250 characters to serve as the documents for the annotation process.

## 2.3 Human-LLM Collaborative Annotation

Crafting a comprehensive long-form summary from various documents is challenging and labor-intensive for human annotators. Meanwhile, LLMs have showcased impressive proficiency in certain annotation tasks (He et al., 2023). With this in mind, we conceptualized a collaborative annotation pipeline of three stages, as illustrated in Figure 2.

**Stage 1: Manual Screening and Information Extraction.** Firstly, human annotators read the query and documents thoroughly. They would extract all useful information from the documents and evaluate its utility. We found over 95% of the queries could be answered by the extracted information, and a few exceptions were discarded.

**Stage 2: LLM-based Candidate Summary Generation.** We leveraged LLMs to construct candidate summaries from the extracted information. Generating summaries from human-extracted content, as opposed to raw documents, avoided the introduction of irrelevant information. We employed ChatGPT[1] in the preliminary annotation phase. As our dataset grew, we fine-tuned an open-source model, ChatGLM2-6B (Du et al., 2022), to provide an extra candidate summary for each sample.

**Stage 3: Manual Refinement and Citation Annotation.** Lastly, human annotators chose the preferred summary among the two LLM-generated candidates, refined its quality, and annotated citations. The chosen summary underwent thorough inspection with non-essential and redundant parts removed. Annotators would cite all supporting documents for verification-worthy sentences, correct groundedness and coherence errors, and supplement missing information. Offering multiple candidate summaries aimed to avoid limiting annotators to a single, potentially lower-quality option and to merge the strengths of different options, thereby improving the quality of the final summary. The core statistics are present in Table 2. More details of the dataset can be found in Appendix A.

[1] We use the gpt-3.5-turbo-0613 checkpoint in this stage.

| **Statistic** | **Value** |
|---|---|
| # Train / Dev / test | 5,630 / 500 / 1,000 |
| # Domains | 16 |
| # Search Results per Query | 5 |
| Full Content Len. / Snippet Len. | 774 / 205 |
| Summary Len. | 167 |
| # Sent. per Summary | 4.56 |
| # Citations / Sub-claims† per Sent. | 1.55 / 1.62 |

Table 2: Core statistics of the WebCiteS dataset. Full content length and snippet length refer to the average length of a single search result (web page). † Sub-claims of sentences are extracted by ChatGPT (Section 4.2).

## 3 Evaluation Framework

The AQFS task targets two dimensions: **summarization utility** and **attribution**. In this section, we introduce the evaluation metrics based on an evaluator with two key components: a **claim-split model** $\psi$ and an **NLI model** $\phi$. The claim-split model decomposes each sentence $s_i$ into a set of sub-claims $\psi(s_i) = \{c_{i,1}, c_{i,2}, \ldots\}$. The NLI model $\phi$ predicts if the given premise entails, contradicts, or is neutral to the given hypothesis. We report the performance of $\phi$ and $\psi$ in Section 4.

### 3.1 Evaluating Summarization Utility

We adopt the following metrics to evaluate the utility of the summaries:

**Length.** We report the average summary length, as prior studies (Gehrmann et al., 2023; Liu et al., 2023c) point out that the summary lengths across different systems exhibit a large variance which is not well-reflected by other metrics.

**Self-BLEU.** Self-BLEU (Zhu et al., 2018) measures the diversity of the generated text. Xu et al. (2023) find this metric effective at evaluating the coherence of long-form answers.

**Claim precision.** We apply $\psi$ to extract all sub-claims from the system summary and calculate the fraction of them being entailed by the reference summary using $\phi$. This metric could measure the *accuracy* and *relevancy* of system summaries.

**Claim recall.** Similarly, we apply $\psi$ to extract all sub-claims from the reference summary and calculate the fraction of them being entailed by the system summary. This metric could measure the *comprehensiveness* of system summaries.

**Claim F$_1$.** Finally, we can compute the claim F$_1$ score of a system by taking the harmonic mean of its claim precision and recall. See Appendix B.1 for more discussions on summarization metrics.

### 3.2 Evaluating Attribution

Only *verification-worthy* sentences with *citation mask* $m_i = 1$ are included in attribution evaluation. Gao et al.'s (2023b) automatic metrics assume that all sentences need citations, i.e. the citation mask $m_i$ is always 1. However, we observe some exceptions such as the introductory or concluding sentences in the summaries (see Section 2.1). Therefore, we propose to automatically predict the citation mask for sentence $s_i$:

$$m_i = \mathbb{1}(\mathcal{C}_i \neq \emptyset \vee \phi(\mathcal{S}^*, s_i) \neq \text{entailment})$$

where $s_i \in \mathcal{S}$, $\mathcal{S}^* = \{s_j | s_j \in \mathcal{S} - \{s_i\}, \mathcal{C}_j \neq \emptyset\}$ Namely, the citation mask $m_i$ is 1 if one of the conditions is satisfied: (1) The citations $\mathcal{C}_i$ of $s_i$ is not empty, as all model-generated citations should be verified; (2) $s_i$ is not entailed by $\mathcal{S}^*$, as we assume introductory or concluding sentences should be entailed by the rest of the summary.[2]

For sentences with citation mask $m_i = 1$, we evaluate two aspects for attribution: **groundedness** and **citation quality** with the following metrics:

**AIS.** The AIS score assesses if the generation is *attributable to identified sources* (Bohnet et al., 2022; Rashkin et al., 2023). We adopt the fine-grained version proposed in RARR (Gao et al., 2023a), which calculates the fraction of attributable sentences in the generation. Since citations serve as the *identified sources* in AQFS, **a sentence** $s_i$ **is attributable if it is fully supported by its citations** $\mathcal{C}_i$. In practice, our automatic evaluator will classify $s_i$ as attributable if (1) $s_i$ does not contradict any citation in $\mathcal{C}_i$ and (2) $s_i$ or **all** of its sub-claims $\psi(s_i)$ are entailed by the citations. Under this definition, the AIS score is equivalent to the *citation recall* metric proposed in ALCE (Gao et al., 2023b). Since this metric generally measures both citation quality and groundedness, we adopt the term *AIS* and define a variant of *citation recall*.

**ACS.** We propose *attributable to contextual sources* (ACS), a variant of the AIS score that uses oracle citations from the evaluator rather than the model-generated citations for evaluation. This isolates groundedness assessment by eliminating the impact of citation errors. For example, if $s_i$ is con-

[2] $\mathcal{S}^*$ only involves sentences with non-empty citations. Without this restriction, the evaluation may be hacked if the model generates two mutually entailed sentences without citations. In this case, both of their citation masks would be 0, making them escape from attribution evaluation.

textually grounded but does not cite any source, its AIS and ACS scores will be 0 and 1 respectively.

**Citation precision.** This metric measures if the sentence is correctly supported by each of its citations. For $s_i$, we extract the model-generated citations $C^i_{\text{pred}}$ and obtain the oracle citations $C^i_{\text{ref}}$ involving all $d_j \in \mathcal{D}$ that **fully or partially support** $s_i$. Then, citation precision for $s_i$ is:

$$\text{Citation Prec}(s_i) = \frac{|C^i_{\text{pred}} \cap C^i_{\text{ref}}|}{|C^i_{\text{pred}}|}$$

In practice, the evaluator would include $d_j$ into $C^i_{\text{ref}}$ if (1) it entails $s_i$ or (2) it does not contradict $s_i$ and entails **any** of its sub-claims. Moreover, if $C^i_{\text{pred}}$ is empty, we seek the nearest non-empty citations $C^{i*}_{\text{Pred}}$ from the subsequent sentences of $s_i$ to replace $C^i_{\text{Pred}}$ for evaluation. This is to avoid penalizing the model for generating multiple sentences based on the same sources but only adding citations in the final sentence. If $C^{i*}_{\text{Pred}}$ is not found, citation precision for $s_i$ woule be zero. Finally, we average the scores of all $s_i \in \mathcal{S}, m_i = 1$ as the citation precision score for the summary $\mathcal{S}$.

**Citation recall.** This metric measures if the sentence comprehensively cites all supporting sources:

$$\text{Citation Rec}(s_i) = \frac{|C^i_{\text{pred}} \cap C^i_{\text{ref}}|}{|C^i_{\text{ref}}|}$$

Similarly, we seek $C^{i*}_{\text{Pred}}$ if $C^i_{\text{pred}}$ is empty. We assign a zero score if $C^i_{\text{Ref}}$ is empty. Finally, we average the citation recall scores of all $s_i \in \mathcal{S}, m_i = 1$.

**Citation F$_1$.** Similar to Claim F$_1$, we compute the citation F$_1$ score of a system by taking the harmonic mean of its citation precision and recall.

Figure 3 shows the framework of attribution evaluation. Compared to Gao et al.'s (2023b) automatic methods, our method (1) considers citation mask, (2) incorporates the claim-split model for partial support detection, and (3) distinguishes groundedness and citation quality. See in Appendix B.2 for more discussions on attribution metrics.

## 4 Evaluating the Automatic Evaluator

In this section, we assess the reliability of the automatic evaluator with $\phi$ and $\psi$.

### 4.1 Performance of the NLI model

We evaluate the performance of different open-source NLI models in predicting human-annotated citations in WebCiteS. We finally choose an mT5 model (Xue et al., 2021) fine-tuned on multilingual NLI tasks as $\phi$ for our evaluator,[3] since it achieves the highest accuracy of 82.3% among all models. See Appendix C for more details in NLI models.

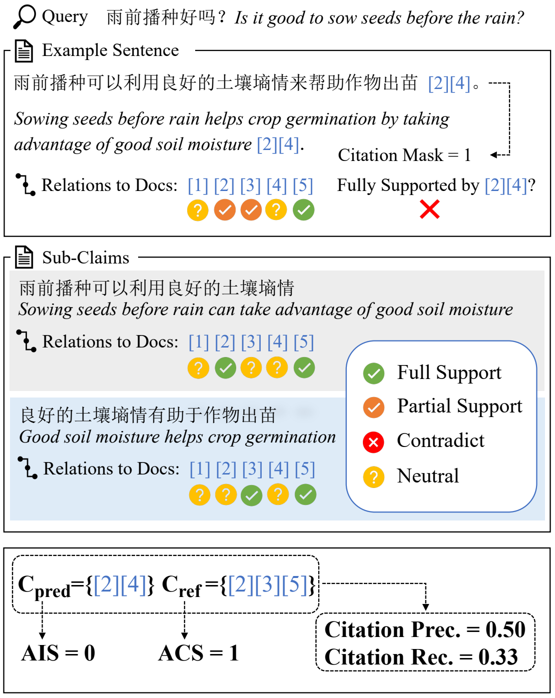

Query 雨前播种好吗？Is it good to sow seeds before the rain?
Example Sentence
雨前播种可以利用良好的土壤墒情来帮助作物出苗 [2][4]。
Sowing seeds before rain helps crop germination by taking advantage of good soil moisture [2][4].
Citation Mask = 1
Relations to Docs: [1] [2] [3] [4] [5]
Fully Supported by [2][4]?
Sub-Claims
雨前播种可以利用良好的土壤墒情
Sowing seeds before rain can take advantage of good soil moisture
Relations to Docs: [1] [2] [3] [4] [5]
良好的土壤墒情有助于作物出苗
Good soil moisture helps crop germination
Relations to Docs: [1] [2] [3] [4] [5]
Full Support
Partial Support
Contradict
Neutral
C_pred={[2][4]} C_ref ={[2][3][5]}
AIS = 0
ACS = 1
Citation Prec. = 0.50
Citation Rec. = 0.33

Figure 3: Illustration of our attribution evaluation. We use a claim-split model to extract sub-claims of a sentence and conduct fine-grained verification on all the source documents. The translation is in italic text.

### 4.2 Performance of the Claim-Split Model

We first prompt ChatGPT[4] to extract the sub-claims in sentences since Kamoi et al. (2023) have validated this approach via human assessment. As using proprietary LLMs for automatic evaluation still faces limitations in efficiency and cost, we additionally fine-tune mT5 on the outputs of ChatGPT to learn this task. We evaluate the claim-split models with the following metrics:

**Redundancy**, which measures if the model generates redundant sub-claims of the source sentence. Two sub-claims $c_i$ and $c_j$ in $\psi(s)$ are deemed redundant if they entail each other: $\phi(c_i, c_j) = 1$ and $\phi(c_j, c_i) = 1$. Based on this, we could eliminate the redundancy of $\psi(s)$: if multiple sub-claims are redundant, we only keep the first one and remove the rest. The resulting set is denoted as $\psi^*(s)$. Fi-

[3] huggingface.co/alan-turing-institute/mt5-large-finetuned-mnli-xtreme-xnli

[4] We use the gpt-3.5-turbo-1106 checkpoint.

| | Redun. ↓ | # Splits | Correct. | Complete. |
|---|---|---|---|---|
| mT5-Base | 2.4 | **1.9** | **99.0** | 90.0 |
| mT5-Large | **1.7** | **1.9** | **99.0** | **92.2** |
| ChatGPT | **1.7** | **1.9** | 96.7 | 89.3 |

Table 3: Performance of different claim-split models.

| Claim-Split | Citation Prec. | Citation Rec. | AIS (HumanCite) | AIS (AutoCite) |
|---|---|---|---|---|
| **Default Citation Mask** | | | | |
| – | 77.7 | 74.5 | 84.8 | 90.3 |
| mT5-Large | 77.7 | 82.3 | 87.3 | 93.1 |
| ChatGPT | 77.9 | 81.0 | 87.1 | 92.7 |
| **Auto Citation Mask** | | | | |
| – | 81.7 | 73.4 | 84.8 | 89.8 |
| mT5-Large | 81.7 | 82.3 | 87.4 | 92.7 |
| ChatGPT | 81.9 | 80.7 | 87.1 | 92.4 |
| **Human Citation Mask** | | | | |
| – | 82.4 | 74.0 | 85.4 | 90.0 |
| mT5-Large | 82.4 | **83.0** | **88.2** | **93.1** |
| ChatGPT | **82.6** | 81.3 | 87.9 | 92.7 |

Table 4: Performance of the automatic evaluator on evaluating the attribution in human-annotated summaries. We use a fixed NLI model and vary the use of claim-split models under different citation mask settings.

nally, the metric is computed as:

$$\text{Redundancy}(\psi(s)) = \frac{|\psi(s)| - |\psi^*(s)|}{|\psi(s)|}$$

**# Splits**, which is the average count of non-redundant sub-claims per sentence. This could reflect the granularity of model outputs, as a lower value indicates that the model may not effectively separate some sub-claims in the source sentence.

**Correctness**, which is the fraction of sub-claims being entailed by the source sentence using the NLI model $\phi$, as we assume that all correct sub-claims should be entailed by the source sentence.

**Completeness**, which is a binary value measuring if the source sentence is entailed by the concatenation of all sub-claims. If not, some essential sub-claims may be missing in the model outputs.

Table 3 shows the evaluation result. We first notice that ChatGPT and the fine-tuned mT5 models are consistent in *# splits* which reflects the decomposition granularity. Moreover, the mT5 models slightly outperform ChatGPT in *correctness* and *completeness*. The primary reason is that ChatGPT occasionally unfollow the task instruction (Zhou et al., 2023). Therefore, we select the fine-tuned mT5-large model as $\psi$ for our evaluator. See Appendix D for more details in claim-split models.

### 4.3 Performance of the Automatic Evaluator

Finally, we assess our automatic evaluator with $\phi$ and $\psi$ on the test set of WebCiteS. We use it to predict the citations for sentences in the reference summaries, and then assess those citations by taking human citations as ground truth. We also compute the AIS scores using both citations. We use the same NLI model $\phi$ and vary the use of $\psi$ to analyze the impact of the claim-split strategy. Besides, we compare different citation mask settings: (1) *default*, which sets $m_i = 1$ for all sentences; (2) *auto*, which automatically predicts $m_i$ (see Section 3.2); (3) *human*, which only sets $m_i = 1$ for sentences with human citations. The results in Table 4 show that: **(1) Claim-split model helps to detect partial support.** Integrating $\psi$ enhances both citation recall and AIS scores, indicating that the citations that only support part of the sub-claims of the sentences are effectively identified; **(2) Accurate citation mask improves the performance of the evaluator.** Using the human citation mask yields the best overall performance, while the auto citation mask achieves better citation precision compared to the default setting. This result emphasizes the necessity of identifying if a sentence requires citations during evaluation.

Moreover, the Cohen's kappa coefficient between the evaluator (using mT5-large as $\psi$ and auto citation mask) and human annotators on whether a sentence should cite a document is 0.6483, which suggests substantial agreement. This further validates the reliability of the automatic evaluator.

## 5 Experiments on the AQFS Task

We evaluate various models on the AQFS task via two methods: **few-shot prompting (FSP)** and **supervised fine-tuning (SFT).** Our prompt for FSP consists of the task instruction and one-shot demonstration, while the SFT prompt removes the demonstration and condenses the instruction for efficiency. For open-source models, we evaluate mT5, ChatGLM2-6B, ChatGLM3-6B (Du et al., 2022), Baichuan2-7B, and Baichuan2-13B (Yang et al., 2023) via both FSP and SFT.[5] For proprietary models, we evaluate ChatGPT and GPT-4 via FSP.[6] See Appendix E.1 for implementation details.

[5] We use the *chat* version of Baichuan2 models.

[6] We use the gpt-3.5-turbo-1106 checkpoint for ChatGPT and the gpt-4-1106-preview checkpoint for GPT-4.

|  |  | Len. | Self-BLEU ↓ | Claim Prec. | Claim Rec. | Claim $F_1$ | Citation Prec. | Citation Rec. | Citation $F_1$ | AIS | ACS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FSP | ChatGPT | 223 | 12.2 | 53.5 | 56.5 | **54.9** | 71.2 | 64.8 | 67.8 | 75.1 | 84.7 |
|  | GPT-4 | 285 | 5.6 | 46.1 | **65.8** | 54.2 | 72.2 | 71.4 | 71.8 | 75.7 | 81.1 |
|  | ChatGLM2-6B | 256 | 11.7 | 40.1 | 45.1 | 42.5 | 9.9 | 7.3 | 8.4 | 9.8 | 73.4 |
|  | ChatGLM3-6B | 252 | 10.0 | 45.6 | 53.7 | 49.3 | 57.0 | 51.4 | 54.1 | 60.0 | 85.0 |
|  | Baichuan2-7B | 188 | 7.6 | 46.9 | 48.4 | 47.6 | 16.3 | 21.6 | 18.6 | 20.8 | 81.0 |
|  | Baichuan2-13B | 200 | 7.8 | 45.4 | 49.6 | 47.4 | 44.4 | 51.0 | 47.4 | 52.3 | 76.2 |
| SFT | mT5-Base | 142 | 19.7 | 47.2 | 35.9 | 40.8 | 47.2 | 35.9 | 37.7 | 40.0 | 82.0 |
|  | mT5-Large | 142 | 15.7 | 52.9 | 40.6 | 45.9 | 50.2 | 51.1 | 50.6 | 56.0 | **90.3** |
|  | ChatGLM2-6B | 156 | 8.7 | 54.6 | 47.6 | 50.9 | 76.3 | 72.5 | 74.4 | 79.4 | 87.1 |
|  | ChatGLM3-6B | 163 | 8.9 | 55.7 | 49.3 | 52.3 | **78.5** | **73.8** | **76.1** | **81.3** | 89.4 |
|  | Baichuan2-7B | 212 | 9.9 | 52.4 | 53.1 | 52.8 | 68.1 | 67.8 | 67.9 | 71.6 | 81.7 |
|  | Baichuan2-13B | 132 | **5.3** | **58.9** | 42.1 | 49.1 | 67.7 | 72.0 | 69.8 | 74.4 | 81.0 |

Table 5: Results of the AQFS task on WebCiteS, where each sample consists of five snippets as documents.

### 5.1 Main Results

We first adopt the default setting where each sample consists of five snippets of web pages as documents. The experimental results are present in Table 5. In general, we observe a large variance of summary lengths across models. For FSP, ChatGPT and GPT-4 outperform all open-source LLMs on both claim $F_1$ scores and citation $F_1$ scores. However, even GPT-4 exhibits unignorable attribution errors: only 72% of its citations are correct and only 71% of supporting documents are cited. Moreover, only 76% of its generated sentences are fully supported by their respective citations, and only 81% of them are grounded in the input context. For SFT, we observe that smaller pre-trained models such as mT5 significantly lag behind open-source LLMs on both summarization utility and citation quality. Although the fine-tuned mT5-Large model achieves the best groundedness (reflected by the highest ACS score of 90.3%), we find it is primarily due to the model's tendency to copy the input text rather than summarize the content pertinent to the query, which leads to the suboptimal claim $F_1$ scores. Our additional findings include:

**Groundedness errors and citation errors coexist across models.** No model achieves a perfect ACS score, indicating the presence of groundedness errors. Moreover, all models' ACS scores exceed the respective AIS scores. Since their gaps are simply brought by citation errors, this finding reveals that even if the model grounds its generation contextually, it could still struggle with accurate citations. This underscores the challenge of explicit attribution in both open-source and proprietary LLMs.

| Source Type | Max Doc Length | # Docs | Claim $F_1$ | Citation $F_1$ | AIS | ACS |
|---|---|---|---|---|---|---|
| **ChatGPT** |  |  |  |  |  |  |
| Snippet | 250 | 5 | **54.9** | 67.8 | 75.1 | 84.7 |
| Full Content | 512 | 11 | 45.9 | 58.2 | 72.2 | 88.9 |
| Full Content | 256 | 20 | 44.9 | 51.3 | 65.2 | 86.5 |
| **ChatGLM3-6B (SFT)** |  |  |  |  |  |  |
| Snippet | 250 | 5 | 52.3 | **76.1** | **81.3** | **89.4** |
| Full Content | 512 | 11 | 43.6 | 51.9 | 65.8 | 88.3 |
| Full Content | 256 | 20 | 41.8 | 42.9 | 56.3 | 84.9 |
| **Baichuan2-7B (SFT)** |  |  |  |  |  |  |
| Snippet | 250 | 5 | 52.8 | 67.9 | 71.6 | 81.7 |
| Full Content | 512 | 11 | 41.3 | 51.3 | 63.7 | 83.2 |
| Full Content | 256 | 20 | 41.1 | 42.8 | 53.9 | 80.5 |

Table 6: Impact of different document settings. Besides the default setting in Table 5, we further evaluate the model performance in summarizing the full content of web pages. We also adopt different chunk sizes (512 and 256) to analyze the impact of document granularity.

**Supervised fine-tuning improves attribution.** Without fine-tuning, all open-source LLMs struggle with accurate citations. However, SFT consistently boosts all attribution metrics and narrows the gaps between the AIS and ACS scores, indicating a simultaneous optimization of both groundedness and citation quality. This finding highlights the potential benefits of involving the AQFS task during instruction-tuning (Zhang et al., 2023a) to enhance attribution in open-source LLMs.

### 5.2 Results in the Long-Context Setting

We further adopt a more challenging long-context setting, where the models are provided with the full content of web pages to summarize. We chunk the web pages into passages with a maximum length of 512 or 256 and assign a unique citation number

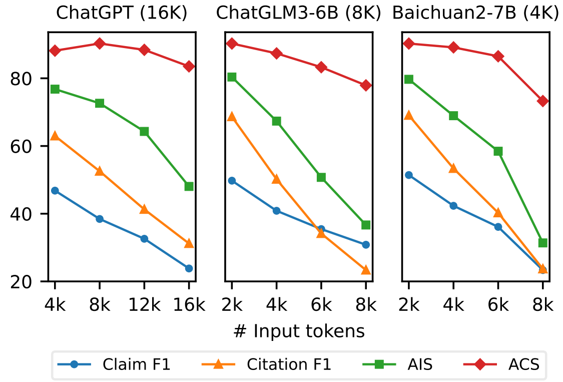


Figure 4: Performance change over context length of the models in Table 6, where full content of web pages are chunked into documents with a maximum length of 512. Model names are followed by their context window size. The number of input tokens is counted using the tokenizer of each model respectively.

to each of them.[7] Table 6 presents the performance of the selected models, which shows that:

**Extending context length reduces model performance.** We observe an overall decline in both summarization utility and attribution with full content instead of snippets. We also visualize the performance variance over the context length in Figure 4, which shows a decline in claim $F_1$, citation $F_1$, and AIS scores as the context length extends. This pattern indicates that extending context length challenges models' ability to synthesize pertinent information and correctly cite sources.

**Using fine-grained documents poses challenges in attribution.** In practice, if the cited documents are too long, it is challenging to verify the model generations, and we expect the models could cite more specific segments of information. Therefore, we also vary the maximum document length when chunking the web pages and investigate the impact of document (citation) granularity. As shown in Table 6, reducing max document length from 512 to 256, without changing the total input evidence, drastically lowers citation $F_1$ and AIS scores for all three models. This performance reduction reveals the difficulty LLMs face in precisely pinpointing the exact supporting evidence within the context.

## 6 Related Work

**Relevant datasets.** Most existing datasets in query-focused multi-document summarization and long-form question answering do not consider attribution in the task setup (Fan et al., 2019; Fabbri et al., 2019; Boni et al., 2021; Stelmakh et al., 2022; Bolotova et al., 2023), while recent initiatives (Qin et al., 2023; Liu et al., 2023b; Gao et al., 2023b) exhibit limitations in citation annotation and evaluation methods detailed in Section 1 and Table 1.

**Evaluating attribution in LLMs.** Attribution refers to the ability to provide external evidence supporting the claims made by the model (Rashkin et al., 2023; Li et al., 2023a). It is crucial for enhancing the credibility of generations and reducing hallucinations (Ji et al., 2023; Zhang et al., 2023c). Although attribution can be approached through various methods, such as generating references from the model's internal knowledge (Weller et al., 2023), retrieval-augmented generation (RAG) with citations (Nakano et al., 2021; Qin et al., 2023; Liu et al., 2023b; Gao et al., 2023b; Li et al., 2023b), and seeking references post-generation (Gao et al., 2023a; Chen et al., 2023; Huo et al., 2023), cost-effective evaluation methods remain a challenging task. Existing automatic metrics (Gao et al., 2023b; Bohnet et al., 2022; Yue et al., 2023) solely depend on off-the-shelf NLI models, failing to detect partial support over complex claims. On the other hand, recent works on summarization evaluation (Liu et al., 2023c), textual entailment (Kamoi et al., 2023), and fact verification (Chen et al., 2023; Min et al., 2023) leverage claim decomposition strategy for fine-grained verification. However, they heavily rely on either human efforts or proprietary LLMs to extract sub-claims, which restricts their scalability. Moreover, existing works do not distinguish the evaluation of groundedness and citation quality. This limits the in-depth understanding of attribution errors in different models.

## 7 Conclusion

In this work, we formulate the task of attributed query-focused summarization (AQFS) and present WebCiteS, a high-quality dataset derived from real-world user queries and search results. We propose a comprehensive evaluation framework on summarization utility and attribution. We also optimize the automatic evaluator to make it better aligned with human citations. Our evaluations on both open-source and proprietary LLMs underscore the unsolved challenge of attribution, especially in the long-context setting with fine-grained documents, highlighting the need for further improvement. We will open-source the dataset and code to facilitate future explorations in this crucial field.

[7] Our evaluator is based on the mT5 model which supports a context window of 512 tokens.

## Limitations

**Task Setup and Dataset.** The AQFS task and the WebCiteS dataset primarily focus on evaluating and improving the model's ability to synthesize information from multiple sources with accurate attribution. Therefore, we do not incorporate retrieval into our task setup. Though we find most web pages returned by the search engine are relevant to the queries (Section 2.3), other works highlight the importance of retrieval quality (Gao et al., 2023b; Liu et al., 2023b). We believe the search results and snippets provided in WebCiteS can also serve as a valuable resource to refine open-source retrievers in future works.

**Evaluation.** The reliability of our automatic evaluator is dependent on the accuracy of both the claim-split model and the NLI model. Though we have validated their performance in Section 4, we could not ensure the model-generated sub-claims are atomic in granularity. Failing to divide sub-claims could affect the identification of partial support instances. Moreover, the context window of the NLI model is also a constraint. Though the mT5 model uses relative position embeddings (Shaw et al., 2018) and accepts arbitrary input sequence length, we find its accuracy drops if the input sequence length significantly exceeds its context window of 512 tokens, primarily due to the length distribution of its training data. Therefore, we believe training a more reliable NLI model for the long-context setting is also an important future work.

## References

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

Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *ArXiv preprint*, abs/2212.08037.

Valeriia Bolotova, Vladislav Blinov, Sofya Filippova, Falk Scholer, and Mark Sanderson. 2023. WikiHowQA: A comprehensive benchmark for multi-document non-factoid question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5291–5314, Toronto, Canada. Association for Computational Linguistics.

Odellia Boni, Guy Feigenblat, Guy Lev, Michal Shmueli-Scheuer, Benjamin Sznajder, and David Konopnicki. 2021. Howsumm: a multi-document summarization dataset derived from wikihow articles. *ArXiv preprint*, abs/2110.03179.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023. Complex claim verification with evidence retrieved in the wild. *ArXiv*, abs/2305.11859.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. 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.

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.

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.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *J. Artif. Int. Res.*, 77.

Xingwei He, Zheng-Wen Lin, Yeyun Gong, Alex Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2023. Annollm: Making large language models to be better crowdsourced annotators. *ArXiv preprint*, abs/2303.16854.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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, page 11–20, New York, NY, USA. Association for Computing Machinery.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Ryo Kamoi, Tanya Goyal, Juan Rodriguez, and Greg Durrett. 2023. WiCE: Real-world entailment for claims in Wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583, Singapore. Association for Computational Linguistics.

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, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

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

Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2023b. Llatrieval: Llm-verified retrieval for verifiable generation.

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 F. Liu, Tianyi Zhang, and Percy Liang. 2023a. Evaluating verifiability in generative search engines. *ArXiv preprint*, abs/2304.09848.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023b. Webglm: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 4549–4560, New York, NY, USA. Association for Computing Machinery.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023c. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training.

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.

Reiichiro Nakano, Jacob Hilton, S. Arun Balaji, Jeff Wu, Ouyang Long, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin

Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv preprint*, abs/2112.09332.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

R. Passonneau. 2009. Formal and functional assessment of the pyramid method for summary content evaluation*. *Natural Language Engineering*, 16:107 – 131.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Yujia Qin, Zihan Cai, Dian Jin, Lan Yan, Shihao Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, Ruobing Xie, Fanchao Qi, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. WebCPM: Interactive web search for Chinese long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8968–8988, Toronto, Canada. Association for Computational Linguistics.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

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*, pages 1–64.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *ArXiv preprint*, abs/1909.08053.

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.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2023. "according to ..." prompting language models improves quoting from pre-training data.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Ai Ming Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Hai Zhao, Hang Xu, Hao Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kuncheng Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Pei Guo, Ruiyang Sun, Zhang Tao, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yan-Bin Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models. *ArXiv*, abs/2309.10305.

Xiang Yue, Boshi Wang, Kai Zhang, Ziru Chen, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. *ArXiv preprint*, abs/2305.06311.

Jiaxing Zhang, Ruyi Gan, Junjie Wang, Yuxiang Zhang, Lin Zhang, Ping Yang, Xinyu Gao, Ziwei Wu, Xiaoqun Dong, Junqing He, Jianheng Zhuo, Qi Yang, Yongfeng Huang, Xiayu Li, Yanghan Wu, Junyu Lu, Xinyu Zhu, Weifeng Chen, Ting Han, Kunhao Pan, Rui Wang, Hao Wang, Xiaojun Wu, Zhongshen Zeng, and Chongpei Chen. 2022. Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence. *ArXiv preprint*, abs/2209.02970.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023a. Instruction tuning for large language models: A survey.

Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori Hashimoto. 2023b. Benchmarking large language models for news summarization. *ArXiv preprint*, abs/2301.13848.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023c. Siren's song in the ai ocean: A survey on hallucination in large language models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv preprint*, abs/2303.18223.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery.

# A The WebCiteS Dataset

Table 10 presents an annotated example in WebCiteS. Table 7 displays the domain distribution of the user queries. Figure 5 displays the distributions of citation numbers. The distribution of context length for using snippets or full content as documents are shown in Figure 6 and Figure 7 respectively.

## A.1 More Details of Data Annotation

**Sample selection.** After 40,000 raw queries were gathered, we adopted a rule-based system to remove common trivial queries such as looking for synonyms, spelling, or translation of a word. We also filter out queries seeking health and medicine advice, since these scenarios are of high risks and hard to judge without professional expertise. Through this process, we obtain 18,500 filtered queries. To ensure the quality of the data, the remaining samples were further filtered by human annotators. Specifically, a sample would be discarded if it matched the following scenarios:

1. If the query was too trivial and did not need long-form answers, or if it was seeking creative inspirations which did not need to be supported by evidence.

2. If the query did not express its demand clearly and was hard to understand.

3. If the query and documents contained inappropriate contents, such as discrimination against specific groups, encouragement of illegal activities, and controversial sentences regarding politics or religions.

4. If the query could not be answered by the retrieved documents.

**Stage 1: manual screening and information extraction.** We developed annotation software which allowed annotators to highlight clause-level segments containing useful information. The manual filtering of invalid samples based on the above criteria also took place in this stage.

**Stage 2: LLM-based candidate summary generation.** In the early annotation phase, We employed ChatGPT to summarize the information extracted from each sample. As our dataset grew, after accumulating 1.2k samples, we additionally fine-tuned an open-source model, ChatGLM2-6B, to provide an extra candidate summary for each sample. We upgraded this model iteratively with the influx of new annotations. Instead of generating multiple outputs by ChatGPT, the motivation for fine-tuning an extra model is to increase the diversity of the candidates.

| Domain | | Count | Domain | | Count |
|---|---|---|---|---|---|
| 生活知识 | Daily Life Knowledge | 1370 | 金融 | Finance | 351 |
| 教育培训 | Education and Training | 1032 | 房产装修 | Real Estate and Decoration | 256 |
| 政策法规 | Policies and Regulations | 726 | 生产制造 | Manufacturing | 221 |
| 商品 | Commodities | 682 | 游戏娱乐 | Gaming and Entertainment | 182 |
| 其他 | Others | 508 | 交通出行 | Transportation | 158 |
| 机动车 | Vehicles | 430 | 旅游 | Travel | 146 |
| 信息技术 | Information Technology | 424 | 母婴育儿 | Maternity and Childcare | 135 |
| 动植物 | Flora and Fauna | 398 | 民俗文化 | Folk Culture | 111 |

Table 7: The domain distribution of the user queries in WebCiteS, covering a broad range of real-world scenarios.

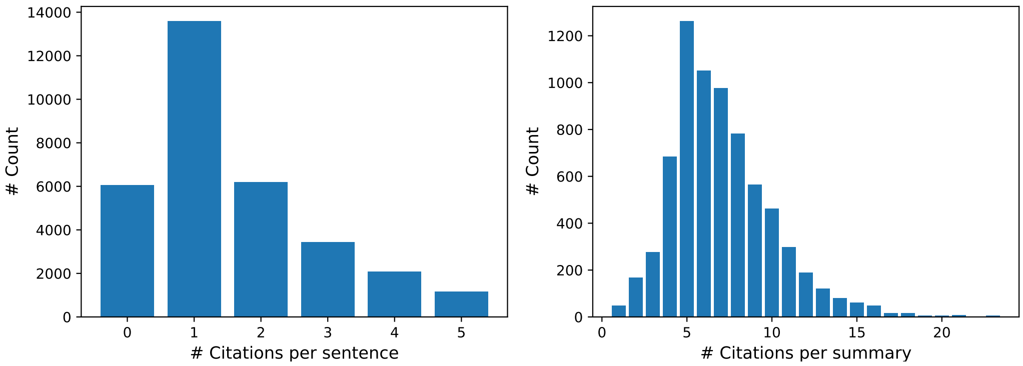


Figure 5: The distribution of the number of citations per sentence and summary in WebCiteS.

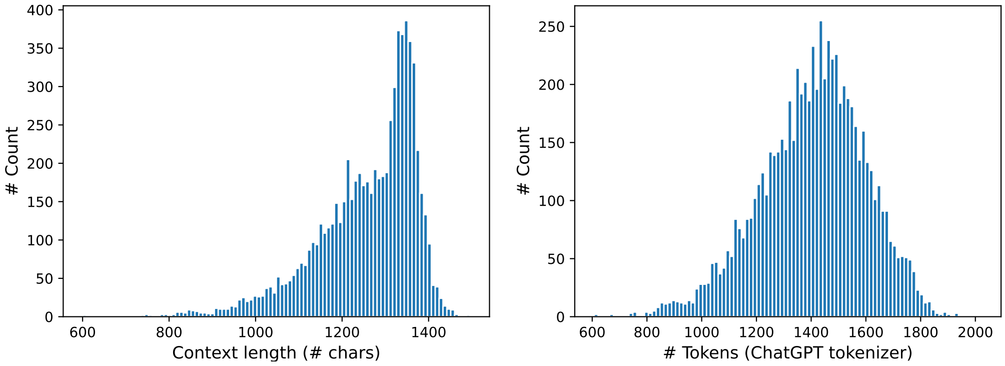


Figure 6: The distribution of the input context length and the number of input tokens in the default setting, where each sample consists of five snippets of web pages as documents. We use the tokenizer of gpt-3.5-turbo.

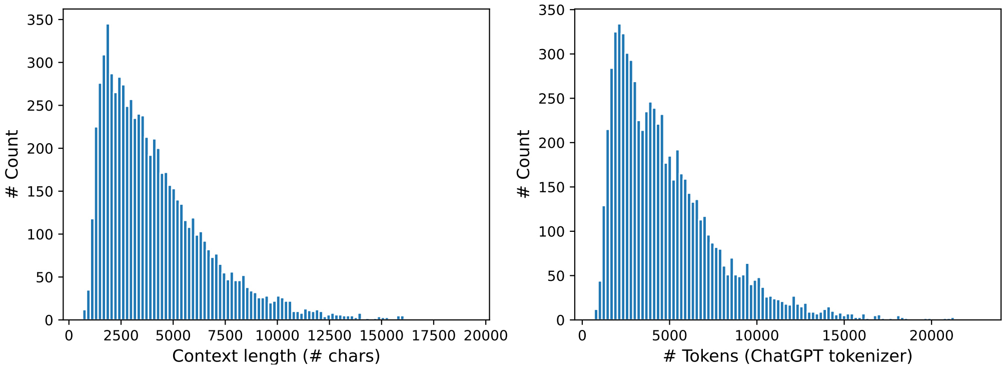


Figure 7: The distribution of the input context length and the number of input tokens in the long-context setting, where we chunk the full content of web pages into documents with a maximum length of 512 characters. We use the tokenizer of gpt-3.5-turbo.

**Stage 3: manual refinement and citation annotation.** We outline a streamlined refinement process as follows: first, annotators were instructed to examine each sentence in the chosen summary, removing unimportant or redundant content. The importance of content was based on the extracted information from the first stage. After that, annotators would identify the verification-worthy sentences in the summary, compare them with all documents with highlighted extraction, and cite all supporting ones. They would also rectify any hallucinations or groundedness errors in the sentences detected during citation annotation. After adding citations, they further ensured all the extracted information was referenced by the summary. If any important information was missing, they would either expand existing sentences or craft new ones to supplement the information, thereby enriching the comprehensiveness of the summary. Finally, annotators inspected the entire summary once again and refined its writing to improve coherence. They were also encouraged to add an introductory sentence to the beginning of the summary to enhance its readability.

**Quality control.** We collaborated with crowdsourcing companies for data labeling. We recruited a team of 27 annotators, 7 quality inspectors, and 1 senior quality inspector, all underwent a month-long training. The quality inspectors reviewed all annotations from the first and third stages, and the senior inspector randomly checked the passed ones. Annotations that failed to meet the standards were returned for corrections and re-inspected.

**More analysis.** We also notice that previous works (Krishna et al., 2021; Lewis et al., 2021; Bolotova et al., 2023) point out a high test-train overlap within the dataset would hinder the models to ground generations in the context. While the queries in WebCiteS are non-repetitive, we additionally examine the URLs of all searched web pages serving as documents in different data splits, and find only 0.3% of the URLs in the train splits exist in validation and test splits, eliminating the concern of high test-train overlap.

# B Evaluation metrics

We bring more details and discussions on the evaluation metrics.

## B.1 Metrics of Summarization Utility

**Length.** It is measured in the number of characters. We remove all citations in the summary before computing length.

**Self-BLEU.** We compute Self-BLEU based on BLEU-4. Our implementation of self-BLEU is based on the sacreBLEU library (Post, 2018).

**Claim $F_1$.** Many prior works in summarization evaluation follow the Pyramid protocol (Nenkova and Passonneau, 2004) to decompose the reference summaries into summary content units (Passonneau, 2009; Shapira et al., 2019; Zhang and Bansal, 2021) or atomic content units (Liu et al., 2023c), and measure how many units are covered in the system summaries (i.e., recall-based metrics). One advantage of recall-based metrics is that only reference summaries need to be decomposed since early works mostly rely on human efforts for sentence decomposition. On the other hand, recent studies also investigate the use of proprietary LLMs (Gao et al., 2023b; Min et al., 2023; Kamoi et al., 2023). For example, Gao et al. (2023b) use InstructGPT (Ouyang et al., 2022) to generate three sub-claims for each reference answer and compute claim recall to measure the correctness of model generations. However, the cost and rate limits of proprietary LLMs still hinder the scalability. In this work, we train a tailored claim-split model that enables us to calculate both claim precision and recall by extracting all sub-claims from both the system summaries and reference summaries with minimum cost. Moreover, we do not use traditional automatic metrics such as ROUGE (Lin, 2004) since their limitations in evaluating LLM generations have been discussed in recent works. (Zhang et al., 2023b; Gao et al., 2023a; Xu et al., 2023).

## B.2 Metrics of Attribution

**Granularity.** We evaluate all metrics of attribution at the sentence level, primarily because the citations in the WebCiteS are annotated at the sentence level, as more fine-grained annotation would require annotators to manually extract sub-claims from a sentence which bring extra cost. However, future works could extend these metrics to the sub-claim level depending on their needs.

**Citation precision.** Liu et al. (2023a); Gao et al. (2023b) compute citation precision by calculating the fraction of accurate citations within the whole

| | Citation Mask | | |
|---|---|---|---|
| | Default | Auto | Human |
| XLM-RoBERTa-Large-XNLI | 75.1 | 78.0 | 78.4 |
| ELS-RoBERTa-Large-NLI | 74.2 | 76.2 | 77.6 |
| ELS-MBERT-1.3B-NLI | 76.0 | 79.1 | 79.3 |
| mT5-Large-XNLI | **78.6** | **82.3** | **82.6** |

Table 8: Performance of citation prediction using different NLI models under three citation mask settings.

generation. The major differences in our approach are:

1. We try to look for $C^{i*}_{\text{Pred}}$ if $C^{i}_{\text{Pred}}$ is empty to avoid unnecessary penalization which leads to the undervaluation of model performance.

2. We compute citation precision at the sentence level and average them for the whole summary, while Liu et al. (2023a); Gao et al. (2023b) compute this metric directly at the response-level.

**Citation recall.** Liu et al. (2023a); Gao et al. (2023b) define citation recall as the fraction of sentences being fully supported by their citations. This is equivalent to the AIS score (Bohnet et al., 2022; Rashkin et al., 2023; Gao et al., 2023a) by taking citations as the *identified sources*. In contrast, our definition of citation recall is consistent with citation precision by calculating the fraction of citations, which is aligned with the naming of the metric.

## C Experiments on NLI Model

We evaluate the performance of different NLI models via a citation prediction task on the test set of WebCiteS: for each sentence in the summary and each given document, we use the NLI model to classify whether the sentence should cite the document, and calculate its accuracy by taking human citations as ground truth. Only sentences with citation mask $m_i = 1$ are considered. We adopt three citation mask settings: *default*, *auto*, and *human*, similar to the experiments in Section 4.3. We select the following NLI models for evaluation: (1) XLM-RoBERTa-Large-XNLI, an XLM-RoBERTa model (Conneau et al., 2020) fine-tuned on multilingual NLI datasets.[8] (2) ELS-RoBERTa-Large-NLI, a Chinese RoBERTa model fine-tuned on several NLI datasets (Zhang et al., 2022).[9] (3) ELS-MBERT-1.3B-NLI, a Chinese model based on the MegatronBERT architecture (Shoeybi et al., 2019), fine-tuned on several NLI datasets (Zhang et al., 2022).[10], (4) mT5-Large-XNLI, an mT5 model (Xue et al., 2021) fine-tuned on multilingual NLI datasets.[11]

The results in Table 8 show that the mT5 model achieves the highest accuracy in predicting citations. Moreover, using the default citation mask (i.e., set $m_i = 1$ for all sentences) lowers accuracy across all models, underscoring the necessity of identifying if a sentence is verification-worthy. Besides, we find that results under auto citation mask and human citation mask are notably similar. This validates the effectiveness of our citation mask prediction method proposed in Section 3.2.

## D Experiments on Claim-Split Model

**Data for training and evaluation.** As described in Section 4.2, our approach involves fine-tuning mT5 models with ChatGPT outputs. We craft a comprehensive prompt with detailed instructions, as shown in Table 13. With ChatGPT's feature of structuring JSON outputs, we prompt it to extract all the sub-claims from sentences within the entire summary in a single response, and then split the response into sentence-level claim-split outputs for training and evaluation. We divide the outputs into training, validation, and test sets aligning with the split of the sample in the WebCiteS dataset. However, since the granularity of sub-claims generated by ChatGPT is not always atomic, we make additional adjustments to the data distribution: we keep all sentences with more than one sub-claim and sample an equal number of sentences with a single sub-claim (either because they are not divisible or because ChatGPT fails to separate their sub-claims). This results in a distribution of 16,158 sentences for training, 2,858 for development, and 1,330 for testing.

**Implementation details.** For fine-tuning, we use the batch size of 64 and the learning rate of 1e-4. We use the AdamW (Loshchilov and Hutter, 2019) optimizer and train the models for 5 epochs. For inference, we use greedy decoding and load the model in half-precision to accelerate evaluation.

[8] https://huggingface.co/joeddav/xlm-roberta-large-xnli

[9] https://huggingface.co/IDEA-CCNL/Erlangshen-Roberta-330M-NLI

[10] https://huggingface.co/IDEA-CCNL/Erlangshen-MegatronBert-1.3B-NLI

[11] https://huggingface.co/alan-turing-institute/mt5-large-finetuned-mnli-xtreme-xnli

| Source | Max Doc | # Docs | Len. | Self- | Claim | | | Citation | | | AIS | ACS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | Length | | | Bleu ↓ | Prec. | Rec. | $F_1$ | Prec. | Rec. | $F_1$ | | |
| | | | | | **ChatGPT** | | | | | | | |
| Snippet | 250 | 5 | 223 | 12.2 | 53.5 | **56.5** | **54.9** | 71.2 | 64.8 | 67.8 | 75.1 | 84.7 |
| Full Content | 512 | 11 | 238 | 10.5 | 42.9 | 49.3 | 45.9 | 63.6 | 53.6 | 58.2 | 72.2 | 88.9 |
| Full Content | 256 | 20 | 245 | 10.6 | 41.2 | 49.3 | 44.9 | 58.2 | 45.9 | 51.3 | 65.2 | 86.5 |
| | | | | | **ChatGLM3-6B (SFT)** | | | | | | | |
| Snippet | 250 | 5 | 163 | **8.9** | **55.7** | 49.3 | 52.3 | **78.5** | **73.8** | **76.1** | **81.3** | **89.4** |
| Full Content | 512 | 11 | 170 | **8.9** | 45.6 | 40.8 | 43.1 | 61.7 | 44.7 | 51.9 | 65.8 | 88.3 |
| Full Content | 256 | 20 | 173 | **8.9** | 44.5 | 39.4 | 41.8 | 53.1 | 35.9 | 42.9 | 56.3 | 84.9 |
| | | | | | **Baichuan2-7B (SFT)** | | | | | | | |
| Snippet | 250 | 5 | 212 | 9.9 | 52.4 | 53.1 | 52.8 | 68.1 | 67.8 | 67.9 | 71.6 | 81.7 |
| Full Content | 512 | 11 | 208 | 10.4 | 42.0 | 40.7 | 41.3 | 58.9 | 45.4 | 51.3 | 63.7 | 83.2 |
| Full Content | 256 | 20 | 208 | 10.4 | 41.3 | 40.9 | 41.1 | 50.4 | 37.3 | 42.8 | 53.9 | 80.5 |

Table 9: Full results of model performance in different document settings shown in Table 6.

## E Experiments on the AQFS Task.

### E.1 Implementation Details

**Few-shot prompting (FSP).** Utilizing the in-context learning abilities of LLMs (Brown et al., 2020), we construct a prompt with four parts:

- **Instruction**: A paragraph that introduces the task and describes specific requirements.
- **Demonstration**: An example with the query, source documents, and human-annotated summary as reference.
- **Sample to Summarize**: The query and source documents that the model needs to summarize.
- **Ending**: An ending statement guiding the model to produce the summary as required.

The full prompt is displayed in Table 10.

**Supervised fine-tuning (SFT).** we also fine-tune open-source models in our experiments. To save GPU memory, we condense the above prompt as the input text by shortening the instruction and removing the demonstration. The condensed instruction is present in Table 11.

For mT5 models, we use the batch size of 64, the learning rate of 1e-4, AdamW as the optimizer, and fine-tune them for 5 epochs. For other open-source LLMs, we use the same batch size and optimizer, while setting the learning rate to 2e-5 and fine-tuning them for 1 epoch, as we find more epochs lead to the rise of validation loss. All open-source LLMs are trained on 8 NVIDIA A100 40G GPUs using Deepspeed ZeRO Stage-3 framework(Rajbhandari et al., 2020). We adopt FP16 mixed precision training (Micikevicius et al., 2018) for ChatGLM2 and ChatGLM3, and BF16 mixed precision training for Baichuan2 models, based on their default configurations.

**Inference.** For ChatGPT and GPT-4, we use the default parameters of the OpenAI Chat API; for open-source models, we follow Gao et al. (2023b) to use Nucleus sampling (Holtzman et al., 2020) with top_p=0.95. We load open-source LLMs in either FP16 or BF16 precision to accelerate inference and save GPU memory.

**Data for the long context setting.** In Section 5.2, we adopt a long-context setting where the models are provided with the full content of web pages to summarize. We chunk the web page into documents with a maximum length of 512 or 256 characters. The chunking is performed at the sentence level, where we try to avoid splitting a single sentence into multiple documents. Web pages shorter than the maximum document length are directly taken as the documents.

给定一个问题和多条搜索结果，你需要准确地理解问题的需求，将搜索结果整理总结成回答，并标注参考来源。请注意以下几点：

1. 你的回答需要简洁清晰，逻辑连贯，内容全面，所有观点都需要被搜索结果中的内容所支持。

2. 你的回答需要引用参考来源。你需要在每句话的结尾标注所参考的搜索结果的编号，放在[]中。你需要全面地引用所有能够支持这句话观点的搜索结果。

3. 你需要将多个搜索结果中相似的观点或信息总结为一句话，并同时标注多个引用。

以下是一个示例:

输入:

**结果[1]**：木瓜吃起来苦什么原因

[开始]木瓜吃起来苦主要是因为木瓜是闷熟的,这种情况还是可以吃的。但是味道方面不是十分尽人意,所以起不到健脾开胃的作用。但是即便是木瓜有一些发苦,它里面的膳食纤维还是存在的,所以摄入人体之后还是可以促进肠道蠕动。依旧可以起到理气顺肠的功效,同时木瓜汁中的维生素c、维生素b、维生素a等多种微量元素的含量没有受到影响。[结束]

**结果[2]**：木瓜为什么吃着很苦

[开始]木瓜吃着很苦可能是正常的现象,因为木瓜的口感是略微苦的,如果是特别苦的情况,就有可能是果肉没有清理干净而导致的。并且木瓜的皮较薄,接近皮蛋果肉,也有可能会出现苦味的情况。日常在吃木瓜的时候,要尽量将果肉清洗干净,也要将果皮和木瓜子清理干净,防止出现口感发苦的情况。[结束]

**结果[3]**：木瓜是苦的是什么回事

[开始]木瓜吃起来本身就有一点苦味,特别是成熟度不够的时候,靠近皮的部位也要苦一些,是正常的,尽管放心食用,且木瓜营养丰富,糖分低是很好的保健水果。木瓜蔷薇科木瓜属,灌木或小乔木,高达5-10米,叶片椭圆卵形或椭圆长圆形,稀倒卵形,长5-8厘米,宽3.5-5.5厘米,叶柄长5-10毫米,微被柔毛,有腺齿;果实长椭圆形,长10-15厘米,暗黄色,木质,味芳香,果梗短。花期4月,果期9-10月。[结束]

**结果[4]**：木瓜为什么吃起来有点苦

[开始] 1、木瓜割伤,因为在木瓜生长过程中割伤表皮。木瓜受伤后继续生长时肉质结构发生变化,所以会感觉到苦。2、闷熟的木瓜,如果选购的是生的时候就摘下来闷熟的木瓜就会是感觉到味苦,被霜打过的木瓜也是苦的。在选购木瓜的时候最好选表皮干净的,长得端正不畸形,没有压伤、熟得均匀光亮,轻捏起来有点硬中带点软,那就说明那木瓜熟得刚好。 3、成熟度不够,木瓜成熟之后果肉柔软,很容易就会碰伤,很不适合运输……[结束]

**结果[5]**：木瓜吃起来有苦味是怎么回事 味道口感苦是什么原因

[开始]为什么木瓜会有苦味 ①没有完全成熟 木瓜出现苦味的原因较多,如果外表皮还是绿色的, 说明木瓜没有完全成熟 , 在煮熟之后依然是青涩带有苦味的 。因此在制作木瓜美食时需要挑选成熟的木瓜,注意其软硬程度以及外表皮的颜色变化。 木瓜成熟之后果肉柔软,很容易碰伤或者出现伤口,不利于运输或者木瓜的贩卖。因此瓜农在采摘的时候都会采摘生的木瓜,没有等到木瓜完全成熟。这时候木瓜的外表皮都是绿色的, 需要放置一段时间……[结束]

问题：木瓜为什么吃着有点苦

输出：

木瓜吃起来有点苦的原因有多种可能。一方面，木瓜本身就有一点苦味，特别是成熟度不够的时候，靠近皮的部位也会苦一些[2][3][5]。另一方面，木瓜可能因为被割伤、闷熟以及被霜打过而感觉到苦味[1][4]。如果吃起来特别苦，则可能是果肉没有清洗干净[2]。因此在制作木瓜美食时需要挑选成熟的木瓜,注意其软硬程度以及外表皮的颜色变化[5]。如果表皮干净,长得端正不畸形,没有压伤、熟得均匀光亮,轻捏起来有点硬中带点软，那就说明那木瓜熟得刚好[4]。

{sample to summarize}

现在，请准确地理解问题，根据搜索结果整理总结出回答，并标注参考来源。

Table 10: The prompt we use for few-shot prompting with full instruction. Table 12 presents the translation.

给定一个问题和多条搜索结果，请根据搜索结果整理总结出回答，并标注参考来源。回答需要简洁清晰、逻辑连贯并严格依照搜索结果。你需要将不同的观点分别总结成多句话，将相似的观点总结成一句话并同时标注多个来源。

*Given a question and multiple search results, please summarize the answer based on the search results and cite the reference sources. Answers need to be concise, clear, logically coherent and strictly based on the search results. You need to summarize claims into multiple sentences, while synthesizing similar claims into one sentence and citing multiple sources at the same time.*

Table 11: The condensed instruction used for supervised fine-tuning. The translation is in italic text.

*Given a question and multiple search results, you need to accurately understand the requirements of the question, organize and summarize the search results into an answer, and annotate the reference sources. Please note the following points:*
*1. Your answer needs to be concise, clear, logically coherent, and comprehensive. All claims must be supported by the content in the search results.*
*2. Your answer needs to cite reference sources. You need to annotate the number of the search result referenced at the end of each sentence, placed in brackets. You need to comprehensively cite all the search results that can support the claim of the sentence.*
*3. You need to synthesize similar claims or information from multiple search results into one sentence and cite multiple references simultaneously.*

*Here is an example:*

*Input:*

> ***Result [1]**: The reason why papaya tastes bitter*
> *[Start] The bitterness of the papaya is mainly because the fruit has been ripened in a stifled condition; it's still edible in this state. However, the flavor may not be very satisfying, so it doesn't serve the function of stimulating the appetite and aiding digestion. Nevertheless, even if the papaya is somewhat bitter, the dietary fiber it contains still exists. Therefore, after intake, it can still promote intestinal movement. It can still be effective in regulating qi and smoothing the intestines, and the content of micronutrients like vitamin C, vitamin B, and vitamin A in papaya juice is not affected. [End]*
> ***Result [2]**: Why does papaya taste bitter*
> *[Start] The papaya tasting slightly bitter might be a normal phenomenon, as the flavor of papaya is naturally a bit bitter. However, if it's excessively bitter, it could be due to the flesh not being cleaned properly. Additionally, since papaya skin is thin and close to the edible flesh, it might also contribute to the bitterness. When consuming papaya, it's important to thoroughly clean the flesh, and remove the skin and seeds to avoid a bitter taste. [End]*
> ***Result [3]**: Why is papaya bitter*
> *[Start] Papaya naturally has a bit of bitterness, especially when it's not fully ripe and near the skin. This is normal, so you can eat it with confidence. Moreover, papaya is a nutritious fruit with low sugar content, making it a great health food. Papaya belongs to the Caricaceae family, a shrub or small tree, growing up to 5-10 meters tall. Its leaves are oval-elliptic or oblong-elliptic, sometimes inversely ovate, measuring 5-8 cm in length and 3.5-5.5 cm in width. The petioles are 5-10 mm long, slightly hairy, with glandular teeth. The fruit is elongated oval, 10-15 cm long, dark yellow, woody, fragrant, with a short stalk. The flowering period is April, and the fruiting period is from September to October. [End]*
> ***Result [4]**: Why does papaya taste a bit bitter*
> *[Start] 1. Papaya cut wounds occur due to cuts in the epidermis during the growth process of the papaya. When a papaya continues to grow after being injured, the texture of its flesh changes, which can cause a bitter taste. 2. Stifled ripe papayas, which are picked when unripe and then ripened in a closed environment, may taste bitter. Papayas affected by frost also tend to be bitter. When buying papaya, it's best to choose ones with clean skin, a proper shape without deformities, no bruises, evenly ripened and shiny, and slightly firm but with a bit of softness when squeezed, indicating that the papaya is just rightly ripe. 3. Insufficient ripeness: After papayas mature, their flesh becomes soft and easily bruised, making them unsuitable for transport...[End]*
> ***Result [5]**: Why does papaya taste bitter, what is the reason for the bitter taste*
> *[Start] Why does papaya have a bitter taste? 1. Not Fully Ripe: One of the main reasons for the bitterness in papaya is its lack of full maturity. If the outer skin is still green, it indicates that the papaya isn't fully ripe, and even after cooking, it will retain a green, bitter taste. Therefore, when preparing papaya dishes, it's important to choose ripe papayas, paying attention to their firmness and the color change of the skin. Once papayas mature, their flesh becomes soft and easily bruised or damaged, making them less suitable for transportation or sale. As a result, farmers often harvest papayas while they are still unripe and green, not waiting for them to fully mature. These green-skinned papayas then need to be left for some time to ripen... [End]*

*Question: Why does papaya taste a bit bitter*

*Output:*

> *There are several possible reasons for papaya tasting a bit bitter. On one hand, papaya naturally has a slight bitterness, especially when not fully ripe, and the areas near the skin tend to be more bitter when the fruit is underripe[2][3][5]. On the other hand, papaya may taste bitter due to being cut, stifled ripe, or affected by frost[1][4]. If the papaya tastes particularly bitter, it could be because the flesh hasn't been cleaned properly[2]. Therefore, when preparing papaya dishes, it's important to choose ripe papayas, paying attention to their firmness and the color changes of the skin[5]. A papaya that is clean-skinned, well-shaped without deformities, free of bruises, evenly ripe and shiny, and slightly firm but with a bit of softness when squeezed is likely just ripe[4].*

*{sample to summarize}*

*Now, please accurately understand the question, summarize the search results, and cite the reference sources.*

Table 12: The translation of the prompt in Table 10.

给定一段由多个句子组成的文本（每句话由<eos>结尾），你需要依次判断每个句子是否包含多个子观点或目标对象，并尝试从中拆分出子句，使得每条子句只包含单个观点或单个目标对象。

你需要保证拆分出的子句语义完整，即使脱离原句也能理解。如果你认为原句无法拆分出子句，就直接返回原句。你需要严格按照JSON格式返回结果。

以下为示例：

输入1：白衣服洗后太阳曝晒或高温熨烫会导致加速发黄。<eos> 浸泡时间过长、洗涤次数的增多、使用不当的洗衣粉或肥皂等因素也会导致白衣服变黄发灰。<eos>
输出1：{
“白衣服洗后太阳曝晒或高温熨烫会导致加速发黄。”：
[“白衣服洗后太阳曝晒会导致加速发黄。”，“白衣服洗后高温熨烫会导致加速发黄。”]
“浸泡时间过长、洗涤次数的增多、使用不当的洗衣粉或肥皂等因素也会导致白衣服变黄发灰。”：
[“浸泡时间过长会导致白衣服变黄发灰。”，“洗涤次数的增多会导致白衣服变黄发灰。”，“使用不当的洗衣粉或肥皂会导致白衣服变黄发灰。”]
}
输入2：观看nba直播的软件有腾讯体育、乐视体育、腾讯视频。<eos>其中，腾讯体育可以提供全球高清的赛事直播，覆盖多类运动。<eos>
输出2：{
“观看nba直播的软件有腾讯体育、乐视体育、腾讯视频。”：
[“观看nba直播的软件有腾讯体育”，“观看nba直播的软件有乐视体育”，“观看nba直播的软件有腾讯视频”]，
“其中，腾讯体育可以提供全球高清的赛事直播，覆盖多类运动。”：
[“腾讯体育可以提供全球高清的赛事直播”，“腾讯体育覆盖多类运动。”]
}

注意按顺序对原文所有的句子进行拆分，不要漏掉任何一句。你需要确保拆分出所有子句都来源于同样的原句。为了保证每个子句的语义完整，你可能需要补充一些描述（如子句的主语等）。同时，如果多个子句间存在不可忽视的逻辑关系（如因果关系、条件关系等），则不应将它们拆分，下面是一个例子：

这句话不应该拆分：如果是硬件或者系统问题导致的摄像头黑屏，建议找专业的人员进行检查。

请严格按照JSON格式返回结果。

*You are given sequence of sentences, each ending with "<eos>". For each sentence, determine if it contains multiple sub-claims or target objects. If so, split it into sub-sentences, ensuring each sub-sentence contains only one claim or target object and is semantically complete on its own. If a sentence cannot be split, return it as is. The results should be strictly in JSON format.*

*For example:*

*Input 1: After washing, white clothes exposed to the sun or ironed at high temperatures tend to yellow faster.<eos> Soaking too long, frequent washing, or using inappropriate laundry detergent or soap can also cause white clothes to turn yellow or gray.<eos>*
*Output 1: {*
*"After washing, white clothes exposed to the sun or ironed at high temperatures tend to yellow faster.":*
*["After washing, white clothes exposed to the sun tend to yellow faster.", "After washing, white clothes ironed at high temperatures tend to yellow faster."]*
*"Soaking too long, frequent washing, or using inappropriate laundry detergent or soap can also cause white clothes to turn yellow or gray.":*
*["Soaking too long can cause white clothes to turn yellow or gray.", "Frequent washing can cause white clothes to turn yellow or gray.", "Using inappropriate laundry detergent or soap can cause white clothes to turn yellow or gray."]*
*}*

*Input 2: Apps for watching NBA live include Tencent Sports, LeSports, and Tencent Video.<eos> Among them, Tencent Sports offers global HD live broadcasts, covering a variety of sports.<eos>*
*Output 2: {*
*"Apps for watching NBA live include Tencent Sports, LeSports, and Tencent Video.":*
*["Apps for watching NBA live include Tencent Sports", "Apps for watching NBA live include LeSports", "Apps for watching NBA live include Tencent Video"],*
*"Among them, Tencent Sports offers global HD live broadcasts, covering a variety of sports.":*
*[" Tencent Sports offers global HD live broadcasts", " Tencent Sports covers a variety of sports."]*
*}*

*Ensure to split all sentences in the order they appear, without missing any. Each sub-sentence should originate from the same original sentence. You may need to add some descriptions to ensure the completeness of each sub-sentence. If there are significant logical connections (like cause-effect or conditional relations) between multiple sub-sentences, do not split them. For example,*

*The following sentence should not be split: If the camera blackout is caused by hardware or system issues, it's recommended to consult a professional.*

*Return results in strict JSON format.*

Table 13: The prompt we used to split sub-claims with ChatGPT. The translation is in italic text.