

# Cited Text Spans for Scientific Citation Text Generation

Xiangci Li Yi-Hui Lee Jessica Ouyang

Department of Computer Science

University of Texas at Dallas

Richardson, TX 75080

lixiangci8@gmail.com,

{yi-hui.lee, Jessica.Ouyang}@UTDallas.edu

## Abstract

An automatic citation generation system aims to concisely and accurately describe the relationship between two scientific articles. To do so, such a system must ground its outputs to the content of the cited paper to avoid non-factual hallucinations. Due to the length of scientific documents, existing abstractive approaches have conditioned only on cited paper *abstracts*. We demonstrate empirically that the abstract is not always the most appropriate input for citation generation and that models trained in this way learn to hallucinate. We propose to condition instead on the *cited text span* (CTS) as an alternative to the abstract. Because manual CTS annotation is extremely time- and labor-intensive, we experiment with distant labeling of candidate CTS sentences, achieving sufficiently strong performance to substitute for expensive human annotations in model training, and we propose a human-in-the-loop, keyword-based CTS retrieval approach that makes generating citation texts grounded in the full text of cited papers both promising and practical.

## 1 Introduction

An automatic citation generation system aims to concisely and accurately describe the relationship between two scientific articles, a citing paper and a cited paper. Due to the rigorous nature of academic research, even large language models (LLMs), such as the GPT family (Brown et al., 2020; OpenAI, 2023), still have to ground their outputs to the input cited papers to avoid non-factual hallucinations.

Given a set of papers to cite, prior extractive approaches (Hoang and Kan, 2010; Hu and Wan, 2014; Chen and Zhuge, 2019; Wang et al., 2019a; Deng et al., 2021) selected salient sentences from the full text of the cited papers. However, existing abstractive approaches (AbuRa’ed et al., 2020; Xing et al., 2020; Ge et al., 2021; Luu et al., 2021;

Chen et al., 2021; Li et al., 2022) generate citation texts conditioned on only cited paper *abstracts*, since the full texts of the papers often exceed the length limit of fine-tuned Transformer-based models (Li and Ouyang, 2022). Li et al. (2022) show that, when human judges are asked to rate the *relevance* of human-written gold citations with respect to the cited paper abstract (i.e. the generation system input), the scores are significantly lower than those of system-generated citations. We hypothesize that this discrepancy is caused by the gold citations referring to content from the cited paper body sections, not just the abstract, and these body sections are not available to either the generation model or the human judges.

Intuitively, the *cited text span* (CTS), the specific span of the cited paper that the gold citation refers to, should be a better alternative to the abstract for grounding a citation. Unfortunately, CTS labels are difficult to obtain. Existing manually-annotated sentence-level CTS datasets (Jaidka et al., 2018, 2019; AbuRa’ed et al., 2020) suffer from two key issues: annotation is extremely labor-intensive, even for domain experts, so the datasets are small; and inter-annotator agreement is relatively low (Cohen’s  $\kappa$  of 0.16~0.52). Further, automatic CTS retrieval systems trained on these datasets give low performance ( $F1 < 0.2$ ) and also use the gold citation text as input, making them inapplicable in a citation generation setting. Given these obstacles, CTS-based citation text generation is under-explored.

In this work, we study scalable CTS retrieval for citation text generation without expensive manual annotations. Our main contributions are as follows:

- We argue that CTS retrieval is an important and under-explored subtask of citation text generation. We demonstrate that citation texts generated using CTS are significantly more accurate and faithful than those using the existing approach of cited paper abstracts.

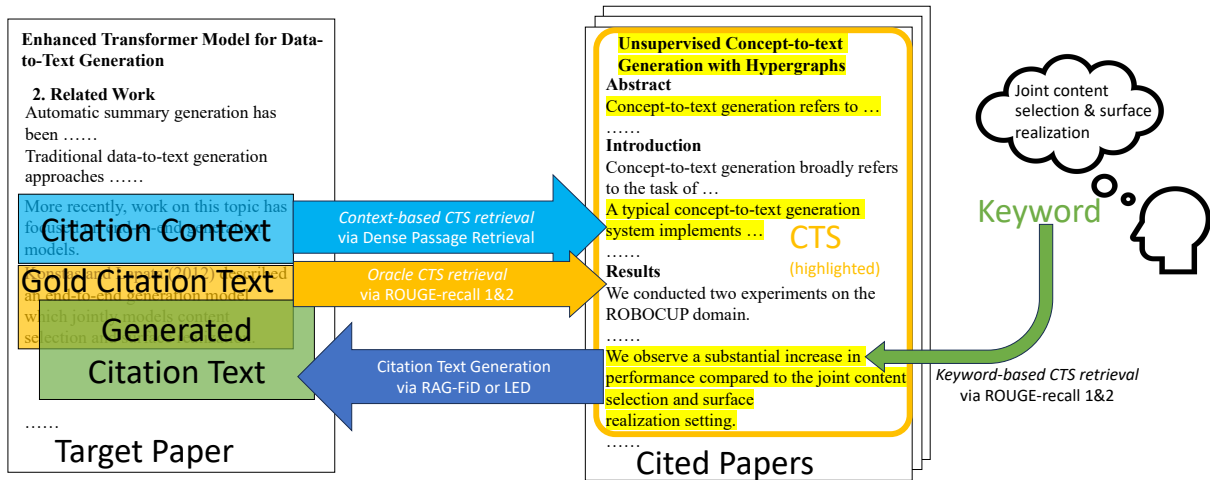


Figure 1: Overview of the proposed CTS-based citation generation approach. The *Context*, *Oracle*, and *Keyword* strategies are used to retrieve CTS from the cited paper (Konstas and Lapata (2012)) and generate a citation text for the target paper (Gong et al. (2019)). See Appendix Table 10 for details of the example.

- Because existing human-annotated CTS datasets are too small to train an effective CTS retrieval system, we show that distant CTS labeling achieves sufficiently good performance to substitute for expensive human annotations. Using this method, we create a large-scale, distantly-labeled CTS dataset.
- We argue that existing CTS retrieval systems are inappropriate for the citation generation task because they use the gold citation text as the retrieval query. Instead, we propose a simple, human-in-the-loop, keyword-based CTS retrieval method that makes generating citations grounded in the full text of the cited paper both promising and practical<sup>1</sup>.

## 2 Background and Related Work

There are two main CTS datasets: Jaidka et al. (2018, 2019) proposed the CL-Scisumm shared task and manually annotated CTS, citation facets, and summaries for 364 citation texts that cite a set of 20 papers; AbuRa’ed et al. (2020) manually annotated sentence-level CTS for 20 related work sections. Unfortunately, both suffer from low inter-annotator agreement, and automatic CTS retrieval systems trained and evaluated on these datasets perform poorly (Jaidka et al., 2018, 2019; AbuRa’ed et al., 2020; Chandrasekaran et al., 2020).

Only a few studies have used CTS for scientific document summarization. ScisummNet (Yasunaga et al., 2019) leveraged a ROUGE-based

CTS retrieval approach (Cao et al., 2015) to collect community-based evidence for summarizing a paper. Wang et al. (2019a) trained a feature-based ensemble model for CTS retrieval using ScisummNet and used the retrieved CTS for extractive related work generation. Syed et al. (2023) used BM25 and SciBERT (Beltagy et al., 2019) to retrieve CTS to provide more contexts for scientific paper summarization. However, all of these works used the target citation text as the query for CTS retrieval, which cannot be used in the citation generation setting that we tackle in this paper.

## 3 A Large-Scale CTS Dataset: Distant vs. Human Labeling

When authors cite a paper, they locate salient passages and compose a citation text grounded in these cited text spans (CTS). Unfortunately, such fine-grained CTS information is not recorded anywhere; even an author may not be able to precisely identify the CTS that they used. While prior CTS retrieval systems have trained on small, human-annotated CTS datasets (Jaidka et al., 2018, 2019; AbuRa’ed et al., 2020; Chandrasekaran et al., 2020), we argue that the relatively low inter-annotator agreement of these datasets suggests there can be multiple reasonable CTS for a given citation, just as there can be multiple valid extractive summaries of a document (see Section 3.3). Thus, distantly-labeled CTS is not necessarily of lower quality than human-annotated CTS, as long as the distant labels identify one of these multiple reasonable candidates. In this section, we use a simple distant approach, with the gold citation text as query,

<sup>1</sup><https://github.com/jacklxc/CTS4CitationTextGeneration>

to automatically label plausible CTS candidates to create a large-scale CTS dataset without expensive and time-consuming human annotations. We demonstrate that our distantly-labeled CTS are of comparable quality to existing human-annotated datasets.

### 3.1 Approach

With the observation that high keyword overlap with the citation text is a crucial characteristic of CTS, we follow Cao et al. (2015); Yasunaga et al. (2019) in using gold citation texts as queries to rank the top- $k$  sentences in the corresponding cited papers as CTS candidates. The similarity is measured by the average of the ROUGE-1 and -2 recall scores (Lin, 2004).<sup>2</sup> We use NLTK<sup>3</sup> to remove stop words for both the query and cited paper sentences. Compared to dense retrievers (Zhang\* et al., 2020), ROUGE-based retrieval does not use GPUs, and thus computationally more efficient. Moreover, since there is no good fine-tuning dataset for CTS retrieval due to the small size of human-labeled datasets Jaidka et al. (2018, 2019), an unsupervised approach is preferable.

We use the human-labeled datasets of Jaidka et al. (2018, 2019) (CL-SciSumm) and AbuRa’ed et al. (2020) as test sets to evaluate our distantly-labeled CTS<sup>4</sup>. We take the union of all annotators’ selected sentences as the human-labeled CTS.

### 3.2 Evaluation Metrics

**Token overlap.** We use ROUGE-recall to measure the coverage of the distantly-labeled tokens that support the gold citation text.

**Faithfulness.** We use QuestEval (Scialom et al., 2021) and ANLI (Nie et al., 2020) to compare the faithfulness of the gold citation text when grounded in human-labeled versus distantly-labeled CTS. For ANLI, we report the non-contradiction score (sum of *entailment* and *neutral*) given by a fine-tuned RoBERTa-large model (Liu et al., 2019).

### 3.3 Findings

**Distant labeling has high coverage of human annotations.** As Figure 2 shows, our top-40 re-

<sup>2</sup>We also experiment with BERTScore (Zhang\* et al., 2020) as a dense retrieval approach, but the latter slightly underperforms ROUGE in all our experiments, including the comparison to human-labeled CTS (Figure 2).

<sup>3</sup><https://www.nltk.org/index.html>

<sup>4</sup>We exclude 45 of AbuRa’ed et al. (2020)’s 239 cited papers due to an XML format error, but we do not expect this omission to affect the overall trend of our evaluation.

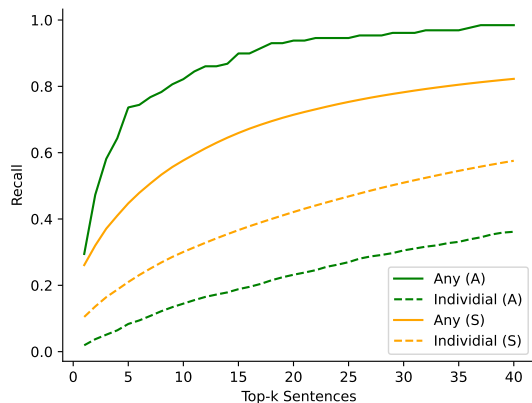


Figure 2: Performance of distantly-labeled CTS, measured by recall against human-labeled CTS. Solid lines (“Any”): each citation counts as one data point; a true positive is when at least one human-labeled sentence is distantly-labeled. Dotted lines (“Individual”): each human-labeled sentence is a separate data point. Green lines (“A”): AbuRa’ed. Yellow lines (“S”): CL-SciSumm.

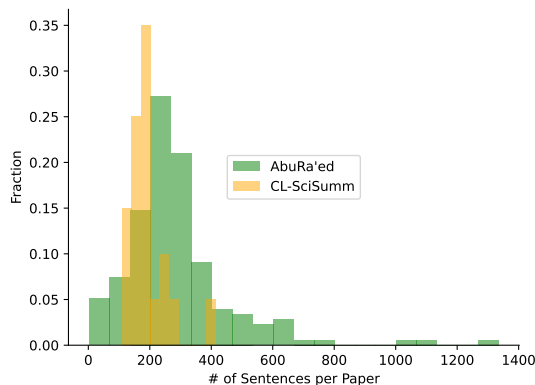


Figure 3: Distribution of the lengths of the cited papers by the number of sentences in CL-SciSumm and AbuRa’ed.

trieved CTS cover 80% and 95% of the human-labeled CTS in CL-SciSumm and AbuRa’ed, respectively. Further, as Figure 3 shows, the papers in these two datasets average 200 sentences long, so we can see that our automatic ranking concentrates the human-annotated CTS into roughly the top 20% (~40/200) of cited paper sentences.

**Distant labeling is more similar to the gold citation text.** As Figure 4 and Table 1 show, although both human-labeled and distantly-labeled CTS are comparably faithful, distantly-labeled CTS has higher token overlap with the target citation text. This is unsurprising, given that using ROUGE as our similarity metric explicitly prioritizes token overlap, while human annotators may not do so. Evaluating human CTS in the intersection, rather than the union, of different annotators’

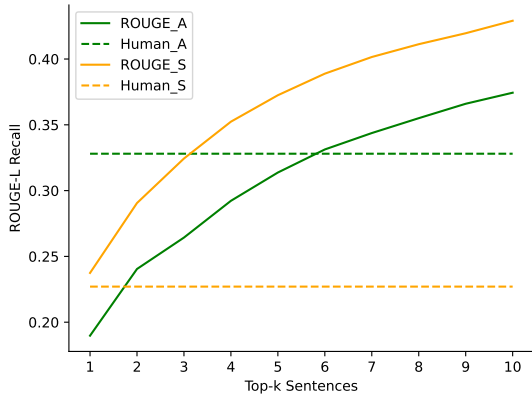


Figure 4: Overlap of the top- $k$  distantly-labeled CTS (solid lines) & human-labeled CTS (dotted lines), measured by ROUGE-L recall against the gold citation text. Green lines (“A”): AbuRa’ed. Yellow lines (“S”): CL-SciSumm. The average length of human-labeled CTS is  $|\bar{A}|=16.8$  and  $|\bar{S}|=2.5$  sentences.

Config	Distantly-Labeled			Human-Labeled		
	R-L	QEval	ANLI	R-L	QEval	ANLI
AbuRa’ed	<b>0.376</b>	0.254	0.959	0.328	0.263	<b>1.000</b>
A-strict	-	-	-	0.224	<b>0.269</b>	0.992
SciSumm	<b>0.429</b>	0.259	<b>0.973</b>	0.225	0.258	0.922
S-strict	-	-	-	0.187	<b>0.275</b>	0.923

Table 1: Distant and human CTS labeling performance measured by ROUGE-L-recall, QuestEval, and ANLI against the gold citation text. “Strict” uses the intersection, rather than union, of human-labeled sentences by different annotators.  $k = 10$  for distant-labeling.

labels (*strict*) has a noticeable impact on the token overlap, indicating that *individual annotators can select different, but all reasonable, CTS sentences*.

**Distant labeling performs better on the downstream generation task.** We also experiment with citation text generation using both distantly-labeled and human-labeled CTS, using the *LED-oracle* model described in Section 4. As Table 2 shows, distantly-labeled CTS performs comparably to human-labeled CTS for the AbuRa’ed dataset and outperforms human labels for CL-SciSumm, confirming that distant CTS labeling produces competitive performance on the downstream generation task. This experiment shows the upper bound of CTS performance using human and distant oracles, demonstrating that *it is not necessary to acquire expensive, human-labeled CTS for training a retriever for downstream citation text generation*.

## 4 Citation Text Generation Using CTS

In this section, we experiment with citation text generation conditioned on CTS, rather than the abstract-only approach used in prior work, and

Config	Distantly-Labeled			Human-Labeled		
	BLEU	METEOR	R-L	BLEU	METEOR	R-L
AbuRa’ed	0.042	0.206	0.220	0.052	0.216	0.201
A-strict	-	-	-	<b>0.058</b>	<b>0.228</b>	<b>0.226</b>
SciSumm	<b>0.064</b>	<b>0.291</b>	<b>0.265</b>	0.061	0.252	0.242
S-strict	-	-	-	0.058	0.236	0.250

Table 2: Citation text generation performance measured by BLEU, METEOR and ROUGE-F1 against the gold citation. “Strict”: intersection, rather than union, of human-labeled sentences.

demonstrate that conditioning on CTS produces more accurate and faithful citations.

### 4.1 Problem Formulation

Following Li et al. (2022), who note the information loss and leak issue of the citation *sentence* generation task, we aim to generate citation *spans* conditioned on (1) the citing paper context and (2) the full text of the cited paper(s) via retrieved CTS. We assume an auto-regressive generation setting, where citations are generated one at a time, in order of their appearance in the citing paper. Thus, we have access to up to two sentences before the target citation to use as context.

**Data.** We use the CORWA citation span<sup>5</sup> dataset Li et al. (2022), which is derived from the ACL partition of S2ORC (Lo et al., 2020). CORWA consists of a human-annotated *training* set and *test* set, as well as an automatically-labeled *distant* training set, with 1654, 1206, and 19784 “dominant”-type citation texts, respectively. We use only the dominant-type citations that are grounded to specific CTS, following Jaidka et al. (2018, 2019); AbuRa’ed et al. (2020), who filter out very short (“reference”-type) citations.

### 4.2 Approach

#### 4.2.1 CTS Retrieval Strategies

**Context-based retrieval.** In this *fully automatic* approach, we retrieve CTS given only the target citation’s context sentences. We apply the distant labeling approach described in Section 3 on the CORWA training set to create a large-scale CTS dataset and use this data to train a Dense Passage Retrieval (DPR; Karpukhin et al., 2020) model. We initialize both the query and candidate document encoders with Aspire (Mysore et al., 2022), a scientific sentence encoder fine-tuned on top of SciBERT (Beltagy et al., 2019). For training, we

<sup>5</sup>To avoid confusion between cited text *spans* and citation *spans*, we will refer to the generation targets as *citation texts*.



use the top-40 distantly-labeled CTS sentences as positive documents and the rest of the sentences in the cited papers as hard-negative documents.

**Unsupervised oracle retrieval.** The distant labeling method described in Section 3 uses the gold citation text as a query; while this approach cannot be used at test time when the target citation text is not known, we use it as an *Oracle* setting to explore the upper-bound of CTS-based citation text generation.

**Human-in-the-loop keyword retrieval.** There may be multiple reasonable citation texts given the same input context, depending on the interests of the reader. Therefore, we also experiment with a practical, human-in-the-loop scenario where a researcher already has a general idea of what they want to know about a particular cited paper, so they input some keywords as a query to guide CTS retrieval. To simulate this scenario, we apply a keyword extractor to the target citation texts and use the extracted keywords as simulated user-input keywords to retrieve CTS using unsupervised retrieval (*Keyword*). See Appendix Table 10 for an example.

To train the keyword extractor, we manually annotate a small set (673 instances) of citation keywords on top of the CORWA dataset. Each instance consists of a short, “reference”-type citation text with manually highlighted keywords<sup>6</sup>. We pre-train a T5-small model (Raffel et al., 2020) on the named-entity recognition labels of the SciREX training set (Jain et al., 2020) and further fine-tune it on manually annotated CORWA keywords as a sequence-to-sequence keyword extractor<sup>7</sup>.

Table 3 shows the token overlap (ROUGE-L recall) between CTS retrieved using each of our three strategies and the target citation, as well as the faithfulness of the target citation grounded in the CTS; we compare against the cited paper abstract, which is used by existing citation generation systems instead of CTS. The token overlap of CTS is much higher than that of the abstract, as we expect, and more importantly, the faithfulness of the target citation grounded in CTS is higher than if it were grounded in the abstract, supporting our hypothesis

<sup>6</sup>We use reference-type citations to train the keyword extractor only, not in any citation text generation experiments.

<sup>7</sup>Alternatively, any reasonable keyword extractor, such as recent LLMs, can also be used to simulate the user-input keywords.

Setting	ROUGE-L	QuestEval	ANLI
Abstract	0.290	0.263	0.868
Context CTS	0.400	0.265	<b>0.964</b>
Oracle CTS	<b>0.533</b>	<b>0.271</b>	0.955
Keyword CTS	0.440	<b>0.271</b>	0.949

Table 3: Quality of citation generation input in terms of token coverage (ROUGE-recall) and faithfulness (QuestEval, ANLI) with respect to the target citation text.

Setting	RAG-FiD			LED		
	BLEU	METEOR	R-L	BLEU	METEOR	R-L
Abstract	0.214	0.397	0.194	0.142	0.380	0.189
Context	0.216	0.397	0.197	0.163	0.359	0.194
Oracle	<b>0.258</b>	<b>0.461</b>	<b>0.292</b>	<b>0.216</b>	<b>0.450</b>	<b>0.312</b>
Keyword	0.239	0.437	0.262	0.182	0.404	0.270

Table 4: Citation text generation performance measured by BLEU, METEOR and ROUGE-F1 against the gold citation. Citation marks are excluded in evaluation.

that the human-written target citations use information from the body sections of the cited papers. Thus, it is important to condition on CTS, not just the cited paper abstract, for citation generation.

## 4.2.2 Citation Text Generation Strategies

**Retrieval-Augmented Generation.** The Retrieval-Augmented Generation (RAG; Lewis et al., 2020b) bridges document retrieval and text generation by fine-tuning the query encoder and generator jointly. RAG uses DPR (Karpukhin et al., 2020) to retrieve  $k$  documents using the input query. We concatenate the target citation’s context sentences and the retrieved CTS sentences<sup>8</sup> as the input to the citation text generator. We use Fusion-in-Decoder (FiD; Izacard and Grave, 2021) with BART-base (Lewis et al., 2020a) as the generator in our experiments.

**Longformer-Encoder-Decoder.** We also experiment with a simple Longformer-Encoder-Decoder (LED; Beltagy et al., 2020) model. We input the concatenation of the target citation context, the citation mark (eg. “Smith et al. (2024)”), and the retrieved CTS for each cited paper.

## 4.2.3 Experimental Settings

**Training.** We train on the large, automatically-labeled CORWA *distant* training set using cross-entropy loss and perform hyper-parameter tuning on the smaller, human-labeled *training* set; details and the hyper-parameter settings are reported in

<sup>8</sup>Unless explicitly specified, to balance performance and GPU memory usage, we retrieve the top  $k = 10$  CTS in all experiments for the inference of citation text generation.

Setting	Fluency	Relevance	Coherence	Overall
Gold	4.71 (4.74)	4.04 (4.18)	<b>4.23 (4.41)</b>	3.87 ( <b>4.16</b> )
Abstract	4.71 (4.72)	4.29 (4.26)	4.06 (3.96)	3.77 (3.63)
Context	4.80 (4.80)	<b>4.41 (4.31)</b>	4.00 (3.96)	<b>3.95 (3.85)</b>
Oracle	4.80 (4.80)	4.18 (4.14)	4.07 (4.09)	3.86 (3.80)
Keyword	<b>4.84 (4.84)</b>	4.21 (4.16)	4.07 (4.07)	3.84 (3.76)

Table 5: Human evaluation with moderate inter-annotator agreement (avg. Kendall’s  $\tau = 0.239$ ). Scores in parentheses are from the second round after the gold citation texts are revealed to the judges.

Setting	RAG-FiD		LED	
	R-L	ANLI	R-L	ANLI
Abstract	0.556	0.745	0.591	0.744
Context	0.641	<b>0.959</b>	0.626	<b>0.954</b>
Oracle	<b>0.726</b>	0.940	<b>0.682</b>	0.950
Keyword	0.630	0.941	0.604	0.946

Table 6: Similarity and faithfulness of the generated citation texts with respect to their input setting, measured by ROUGE-recall and ANLI.

Appendix A. Each model takes about one day to train on one Nvidia Tesla V100S-32GB GPU.

**Testing.** For all experiments, we report performance on the human-labeled CORWA *test* set.

### 4.3 Human Evaluation

Following the setup used in prior works (Xing et al., 2020; Ge et al., 2021; Li et al., 2022), we recruit six NLP graduate students who are fluent in English as human evaluation judges. To increase their coverage of the test set, we split the judges into two groups of three and assign each group 25 instances sampled from the output of the RAG-FiD model. Each instance consists of a single target citation context and several candidate generations: one conditioned on the *Abstract* only and one for each of our three CTS retrieval strategies: *Context*, *Oracle*, and *Keyword*. The target citation context, cited paper abstracts, and retrieved CTS from all three strategies are also provided to the judges. Each generated citation is rated on a five-point Likert scale with respect to *Fluency*, *Relevance* (to the input; either abstract or CTS), *Coherence* (with respect to the context), and *Overall Quality*.

We perform two rounds of evaluation: first, the gold citation texts are mixed in with the candidate generations but are not identified to the judges; second, we reveal the gold citation texts and ask the judges to revise their ratings with this new information.

### 4.4 Results and Discussion

**CTS-based generation outperforms abstract-only generation.** Table 4 shows that the *Oracle*

and *Keyword* CTS settings massively outperform the *Abstract* approach. Moreover, Table 5 shows that *Abstract* receives the lowest human *Overall* rating. In contrast with prior works in the abstract-only setting (AbuRa’ed et al., 2020; Xing et al., 2020; Ge et al., 2021; Luu et al., 2021; Chen et al., 2021; Li et al., 2022), all settings now score well on *Relevance* (higher than 4). Previously, even the gold citation text scored poorly on *Relevance* because the judges were only shown the abstract, which did not contain sufficient information to support the citation texts; our judges are shown the retrieved CTS sentences, which are a much better match for the citation texts.

**CTS-based generation extracts more short phrases.** The ROUGE-recall scores in Table 6 show that, compared to *Abstract*, CTS-based generations are more similar to their inputs, suggesting that models trained on CTS are more extractive. To investigate further, we use Grusky et al. (2018)’s *coverage* and *density* metrics. *Coverage* measures how much of the generated citation text is extracted from the input (abstract or CTS), while *density* measures the size of the extractions (e.g. individual unigrams versus entire sentences). Additionally, we calculate the compression ratio between the lengths of the inputs and the generated citation texts.

As Figure 5 shows, although the *Abstract* setting has the lowest compression ratio, it has a significantly lower *coverage* score than the CTS settings, confirming that the latter is more extractive. From this finding, along with the lower faithfulness scores in Table 6<sup>9</sup>, we can infer that *Abstract* produces more non-factual hallucinations that distort the ideas of the cited paper. This finding supports our hypothesis that, in the abstract-only setting, *the citation generation input does not contain all of the information present in the target gold citation text, forcing the model to learn to hallucinate that missing information*.

Moreover, *Abstract* has a significantly higher *density* score, indicating longer copied sequences from the input, while the CTS-based settings extract and reorganize shorter phrases from the input. Since copying long text spans from other documents results in plagiarism in the citation generation setting, CTS-based generation is again preferable. Nonetheless, we stress that our CTS-

<sup>9</sup>We exclude QuestEval from this table due to an issue arising from the relative compression ratios of the different input settings.

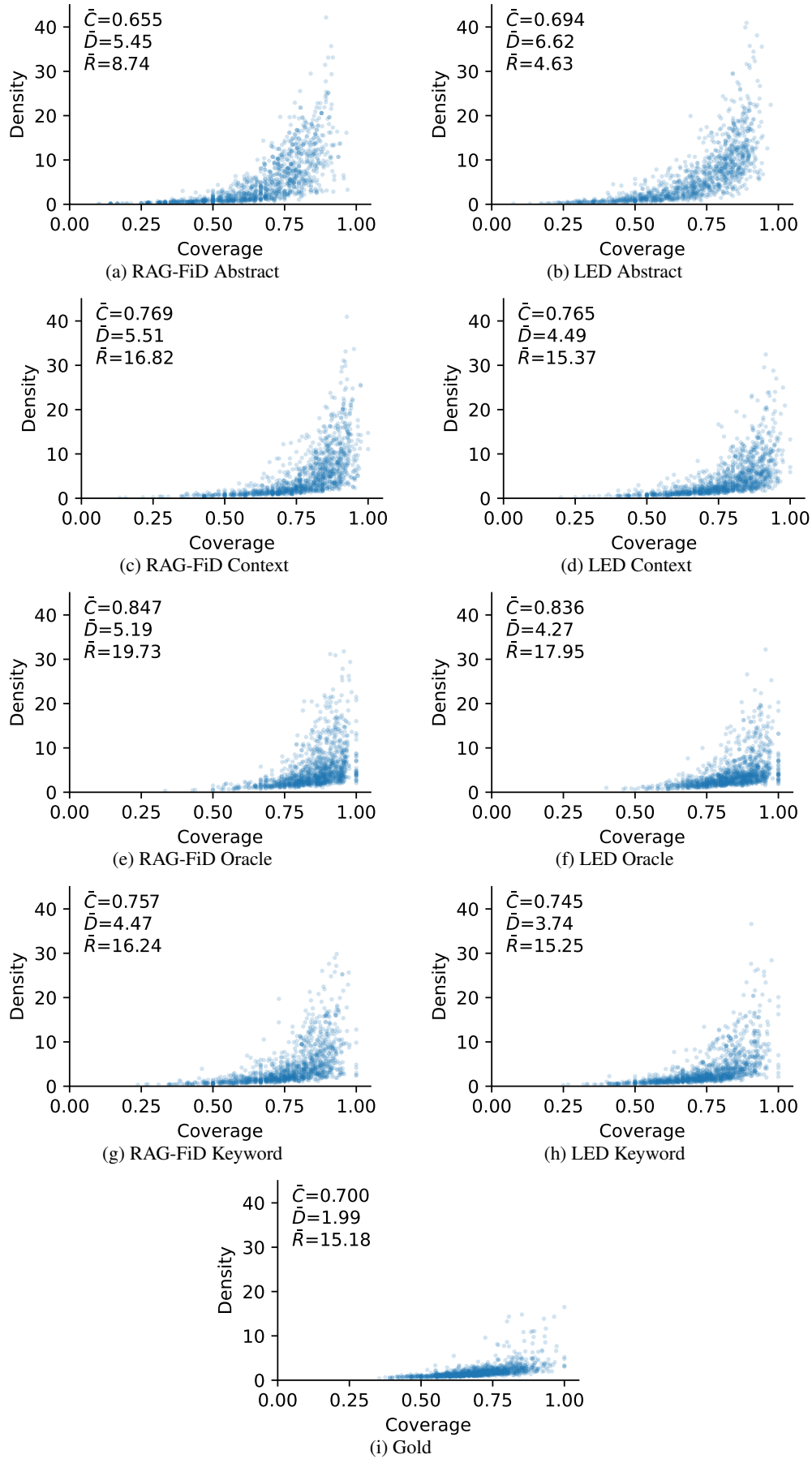


Figure 5: Extractiveness of the generated citation texts, measured by *coverage* and *density* Grusky et al. (2018) against the generation input (abstract or CTS). The average coverage ( $\bar{C}$ ), density ( $\bar{D}$ ), and compression ratio ( $\bar{R}$ ) are shown for each generation setting. The “Gold” setting (Sub-figure 5i) is measured using Oracle CTS as the “input”.

Section	Oracle	Keyword	Context
introduction	2155 (1)	1681 (1)	4140 (1)
related work	1217 (2)	634 (5)	401 (4)
model or architecture	903 (3)	1036 (2)	383 (5)
abstract	712 (4)	816 (3)	3667 (2)
experimental results	710 (5)	703 (4)	92 (8)
conclusion	508 (6)	589 (6)	316 (6)
approach	312 (7)	312 (8)	141 (7)
data	307 (8)	370 (7)	75 (9)
evaluation	210 (9)	215 (10)	38 (12)
discussion	88 (10)	75 (11)	15 (16)
title	73 (11)	262 (9)	921 (3)
background	67 (12)	74 (12)	54 (10)

Table 7: Counts (and ranks) of the most common section headers of retrieved CTS for the CORWA test set. Approximately 12,000 CTS are retrieved in total.

based generation approach is intended to increase accuracy and faithfulness and does not explicitly address the plagiarism issue; additional post-processing, such as a rewriting module, should be used to further reduce the risk of plagiarism.

**Predicting CTS fully automatically is challenging.** Although the fully automatic *Context* setting has strong input retrieval performance (Table 3), its performance on the downstream citation text generation task is mixed. It underperforms on automated metrics (Table 4), but human judges rate it slightly higher than *Oracle* and *Keyword* (Table 5). Table 7 shows that *Oracle* and *Keyword* tend to retrieve CTS from more specific sections, such as “Model or Architecture” or “Results” (Appendix Table 10), suggesting that the human judges may have felt that the generated citation texts contained too much unnecessary detail.

Fully automatic CTS retrieval given only the target citation context sentences is challenging because the citation context and the CTS candidates come from different papers (citing versus cited), and the semantic mismatch between them lowers the performance of the DPR module. In the example in Appendix Table 10, the citation context only mentions “end-to-end generation models”, omitting the “content selection” and “surface realization” keywords that appear in the gold citation text, making CTS retrieval in *context* setting much harder than in *oracle* and *keyword*, which have explicit guidance with all important keywords from the gold citation text.

Interestingly, Table 5 shows that human evaluators cannot distinguish between gold and generated citation texts during the blind evaluation round. When the gold citation text is identified to the judges, they revise its scores higher, while

lowering the scores of the generated candidates. This finding suggests that even humans need some guiding information to help them process citation texts. Thus, as Li et al. (2022) argue, the citation text generation task should be human-in-the-loop, leveraging a researcher’s expertise to reduce both the search space size and cascading error propagation in the pipeline, and our *Keyword*-based CTS retrieval approach provides a practical workaround for the challenge of predicting CTS fully automatically.

## 5 Conclusion

In this work, we show that CTS-based citation text generation is more accurate and faithful than the abstract-based approaches used in prior work. We address the challenges that previously prevented CTS from being used effectively for citation text generation. Distant labeling, such as ROUGE-based labeling, achieves sufficiently strong performance to substitute for expensive manual CTS annotations, allowing us to build a dataset large enough to train a supervised CTS retriever. To avoid the pitfall of using the gold citation text as the retrieval query, which has been used in prior work but would not be possible in a real application, we propose a human-in-the-loop, keyword-based CTS retrieval approach that achieves strong performance in a practical and realistic setting. Finally, our experimental results show that, while retrieving CTS using only the citation context sentences is challenging and does not achieve the best performance using automated metrics, it is preferred by human judges and is a good direction for further investigation.

## Limitations

**Challenge of CTS retrieval.** As discussed in Section 2, existing CTS retrieval approaches all use the gold citation text as the query. Our results also show that, while CTS retrieved using the gold citation text (or its keywords) yields strong generation performance, CTS retrieved with only the citation context sentences is not satisfactory. Therefore, we argue that citation text generation should be divided into two parts: support document retrieval and document-based citation text generation. Based on our results, we can achieve strong generation performance, but the automated retrieval part requires further investigation.

As a workaround solution without a strong,



context-based CTS retriever, we show that, when the user provides a few keywords to guide CTS retrieval, our models can perform quite well. This setting is practical under the human-in-the-loop configuration suggested by Li et al. (2022). However, we realize that this human-in-the-loop setting might be unfriendly to less-experienced researchers who are not sure what keyword to use. Future work may automatically predict the keywords of the gold citation text by performing paragraph- or document-level topic analysis using LLMs.

### Choice of ROUGE-labeling approach & SOTA.

We choose ROUGE-based labeling for CTS because it empirically shows the advantage of using even distantly labeled CTS for citation text generation, compared to using only cited paper abstracts. However, we do not claim that our RAG model using this retriever trained on these distant labels is SOTA. ROUGE-based labeling selects CTS with high lexical overlap with the target citation and may miss CTS that are semantically relevant but less lexically similar. In addition, there are many alternative ways to obtain better performance, such as using a more carefully crafted CTS retriever and training a larger generator with extensive hyperparameter tuning. Instead, we simply claim that using CTS can improve citation text generation performance over using cited paper abstracts, and human labeling of CTS is not required because distantly labeled CTS is of comparable quality.

**Quality Assurance of CTS.** Due to the high cost of manually collecting or evaluating the quality of the retrieved CTS, we are not able to perform quality assurance of the distantly retrieved CTS on a large scale to avoid error-propagation to generated citation texts. Future work can incorporate a fact-checking module to further filter out CTS sentences that do not entail the citation text.

### Ethics Statement

In this work, we focus on showing that retrieved CTS provides relevant support information to enable faithful citation text generation. Unsurprisingly, the trained CTS-based citation text generation models turn out to be more extractive (Section 4.4), directly copying parts of the input CTS to the output citation texts. This copying behavior has the potential to result in plagiarism if the output of our CTS-based citation text generator is directly without further modification. As we mention in Section

4.4, it may be necessary to apply a post-processing step, such as a rewriting module, to avoid plagiarism, which is out of the scope of this work and left for future investigation. We also point out that the existing, abstract-based approach to citation text generation extracts longer subsequences from its input, compared to our CTS-based approach, which extracts shorter subsequences and rearranges their order, so the plagiarism problem is not unique to CTS.

### References

- Ahmed AbuRa'ed, Horacio Saggion, and Luis Chiruzzo. 2020. [A multi-level annotated corpus of scientific papers for scientific document summarization and cross-document relation discovery](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6672–6679, Marseille, France. European Language Resources Association.
- Ahmed AbuRa'ed, Horacio Saggion, Alexander Shvets, and Àlex Bravo. 2020. Automatic related work section generation: experiments in scientific document abstracting. *Scientometrics*, 125:3159–3185.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- 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. *Advances in neural information processing systems*, 33:1877–1901.
- Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.
- Muthu Kumar Chandrasekaran, Guy Feigenblat, Edward Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview and insights from the shared tasks at scholarly document processing 2020: Cl-scisumm, laysumm and longsumm. In *Proceedings of the first workshop on scholarly document processing*, pages 214–224.
- Jingqiang Chen and Hai Zhuge. 2019. Automatic generation of related work through summarizing citations.

- Concurrency and Computation: Practice and Experience*, 31(3):e4261.
- Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xi-angliang Zhang, Dongyan Zhao, and Rui Yan. 2021. [Capturing relations between scientific papers: An abstractive model for related work section generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6068–6077, Online. Association for Computational Linguistics.
- Zekun Deng, Zixin Zeng, Weiye Gu, Jiawen Ji, and Bolin Hua. 2021. Automatic related work section generation by sentence extraction and reordering. In *AII@ iConference*, pages 101–110.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yubin Ge, Ly Dinh, Xiaofeng Liu, Jinsong Su, Ziyao Lu, Ante Wang, and Jana Diesner. 2021. [BACO: A background knowledge- and content-based framework for citing sentence generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1466–1478, Online. Association for Computational Linguistics.
- Li Gong, Josep Crego, and Jean Senellart. 2019. [Enhanced transformer model for data-to-text generation](#). In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 148–156, Hong Kong. Association for Computational Linguistics.
- Max Grusky, Mor Naaman, and Yoav Artzi. 2018. [Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.
- Cong Duy Vu Hoang and Min-Yen Kan. 2010. [Towards automated related work summarization](#). In *Coling 2010: Posters*, pages 427–435, Beijing, China. Coling 2010 Organizing Committee.
- Yue Hu and Xiaojun Wan. 2014. [Automatic generation of related work sections in scientific papers: An optimization approach](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633, Doha, Qatar. Association for Computational Linguistics.
- Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2018. Insights from cl-scisumm 2016: the faceted scientific document summarization shared task. *International Journal on Digital Libraries*, 19(2):163–171.
- Kokil Jaidka, Michihiro Yasunaga, Muthu Kumar Chandrasekaran, Dragomir Radev, and Min-Yen Kan. 2019. The cl-scisumm shared task 2018: Results and key insights. *arXiv preprint arXiv:1909.00764*.
- Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. [SciREX: A challenge dataset for document-level information extraction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Ioannis Konstas and Mirella Lapata. 2012. [Unsupervised concept-to-text generation with hypergraphs](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 752–761, Montréal, Canada. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 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. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

- Xiangci Li, Biswadip Mandal, and Jessica Ouyang. 2022. [CORWA: A citation-oriented related work annotation dataset](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5426–5440, Seattle, United States. Association for Computational Linguistics.
- Xiangci Li and Jessica Ouyang. 2022. Automatic related work generation: A meta study. *arXiv preprint arXiv:2201.01880*.
- 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.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. [S2ORC: The semantic scholar open research corpus](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.
- Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. 2021. [Explaining relationships between scientific documents](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2130–2144, Online. Association for Computational Linguistics.
- Sheshera Mysore, Arman Cohan, and Tom Hope. 2022. [Multi-vector models with textual guidance for fine-grained scientific document similarity](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4453–4470, Seattle, United States. Association for Computational Linguistics.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. [Adversarial NLI: A new benchmark for natural language understanding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- OpenAI. 2023. [Gpt-4 technical report](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. [QuestEval: Summarization asks for fact-based evaluation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. 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.
- Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. [Syntactic scaffolds for semantic structures](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3772–3782, Brussels, Belgium. Association for Computational Linguistics.
- Shahbaz Syed, Ahmad Hakimi, Khalid Al-Khatib, and Martin Potthast. 2023. [Citance-contextualized summarization of scientific papers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8551–8568, Singapore. Association for Computational Linguistics.
- Pancheng Wang, Shasha Li, Haifang Zhou, Jintao Tang, and Ting Wang. 2019a. Toc-rwg: Explore the combination of topic model and citation information for automatic related work generation. *IEEE Access*, 8:13043–13055.
- Yufei Wang, Mark Johnson, Stephen Wan, Yifang Sun, and Wei Wang. 2019b. [How to best use syntax in semantic role labelling](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5338–5343, Florence, Italy. Association for Computational Linguistics.
- Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. 2020. [Automatic generation of citation texts in scholarly papers: A pilot study](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6181–6190, Online. Association for Computational Linguistics.
- Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7386–7393.
- Tianyi Zhang\*, Varsha Kishore\*, Felix Wu\*, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.



Hyper-parameter	Search Range	Used
Learning rate	1e-5 ~ 5e-4	5e-5
RAG-FiD batch size	1 ~ 3	2
LED batch size	-	1
Epochs	2 ~ 10	5
RAG-FiD max sequence length	-	350
LED max sequence length	-	2048
Top $k$ ROUGE-retrieval	1 ~ 20	10

Table 8: Hyper-parameter settings for CTS-based citation text generation.

## A More Implementation Details

### A.1 Opting Out of FAISS

The original RAG implementation was designed for Wikipedia-based dialogue response generation, where every Wikipedia article is a candidate for retrieval. Therefore, FAISS (Johnson et al., 2019) was used to speed up the search process. However, our search space is limited to cited paper sentences, and we need to retrieve CTS separately for each citation and its corresponding cited paper. Unfortunately, FAISS does not support partitioned search, so we replace FAISS with our simpler CTS retrieval strategies, as described in Section 4.2.1.

### A.2 Freezing Query Encoder Weights

The RAG model retrieves candidate documents on the fly while jointly fine-tuning the pretrained DPR query encoder and the generator. In our preliminary experiments on retrieving CTS at the paragraph level, we did not find any significant improvement from fine-tuning the query encoder over simply freezing its weights. Further, due to the much larger candidate search space of the sentence-level CTS setting, the on-the-fly retrieval needed to update the query encoder becomes intractable. Therefore, in our citation text generation experiments, we retrieve and cache CTS sentences for all target citation/cited paper pairs in advance and do not fine-tune the query encoder.

### A.3 Citation Text Generation Hyper-parameter Settings

Table 8 shows the tuned hyper-parameter settings for our CTS-based citation text generation experiments.

## B More Related Work

### B.1 Retrieval and Generation

Our proposed framework of retrieving useful information (CTS) to guide generation is partly inspired by work in knowledge-augmented, open-domain

Setting	RAG-FiD			LED		
	R-L	QEval	ANLI	R-L	QEval	ANLI
Abstract	0.556	<b>0.314</b>	0.745	0.591	<b>0.345</b>	0.744
Oracle	<b>0.726</b>	0.288	0.940	<b>0.682</b>	0.300	0.950
Keyword	0.630	0.297	0.941	0.604	0.304	0.946
Context	0.641	0.310	<b>0.959</b>	0.626	0.315	<b>0.954</b>

Table 9: Similarity and faithfulness of the generated citation texts with respect to their input setting, measured by ROUGE-recall, QuestEval, and ANLI.

dialogue generation, where the combination of Dense Passage Retrieval (DPR; Karpukhin et al., 2020), Retrieval-Augmented Generation (RAG; Lewis et al., 2020b) and Fusion-in-Decoder (FiD; Izacard and Grave, 2021) has been shown to yield strong performance (Shuster et al., 2021).

DPR (Karpukhin et al., 2020) separately applies two BERT-like models (Devlin et al., 2019) as query and document encoders. The DPR model is trained with contrastive loss and uses maximum inner product search (MIPS) to retrieve the documents with the highest cosine similarity between the  $[CLS]$  token representations of the query and the knowledge base documents.

Once informative documents have been retrieved, the RAG (Lewis et al., 2020b) paradigm can be applied using any sequence-to-sequence model. The conditional generation input is paired and encoded with each retrieved document in parallel, and the generation output is jointly decoded using all input/retrieved document pairs. RAG-sequence decodes the entire output sequence based on a fixed distribution over the retrieved documents, while RAG-token decodes each token using a different distribution over the retrieved documents. Finally, Shuster et al. (2021) proposed FiD (Izacard and Grave, 2021) as the decoder in the RAG framework by leveraging the attention mechanism to jointly decode from the retrieved document encodings.

## C QuestEval and CTS-based Citation Text Generation

QuestEval (Scialom et al., 2021) is a question-answering-based faithfulness metric composed of a precision score, where questions are generated from the summary (citation text) and answered using the document (generation input), and a recall score, where questions are generated from the input and answered using the citation text. Table 9 shows the full faithfulness evaluation results for our citation text generation task, including QuestEval.

We find that QuestEval encounters some diffi-



culty evaluating CTS-based citation texts because they are very short (Figure 5 shows that CTS-based citation texts have a much higher compression ratio than abstract-only citation texts); as a result, the QuestEval recall score is extremely low because many questions can be generated from the detailed information in the CTS, but the citation text is too short to contain all of the corresponding answers.

Table 11 shows an example of an *Oracle* CTS citation text that receives a low QuestEval score: Wang et al. (2019b) citing Swayamdipta et al. (2018). We can see that the retrieved CTS sentences include multiple entities and other detailed information that trigger QuestEval to generate many questions that the generated citation text cannot answer. However, a human reader can tell that the *Oracle* citation text is indeed factual; its claim about parse tree validity is supported by Swayamdipta et al. (2018)’s Introduction section: “*Since the scaffold task is not an end in itself, we relax the syntactic parsing problem to a collection of independent span-level predictions, with no constraint that they form a valid parse tree.*”

We can also see that the *Abstract*-only citation text is longer and contains more entities than the *Oracle* citation text, while the *Abstract* input is much shorter and contains fewer entities than the *Oracle* input, so it does not suffer from the same low QuestEval recall score. We find many similar examples in our test set of longer *Abstract* citations being scored higher than shorter *Oracle* citations by QuestEval.

<b>Citation context</b>		
More recently, work on this topic has focused on end-to-end generation models.		
<b>Citation texts</b>		
<i>Abstract</i>	Konstas and Lapata (2012) proposed a joint model that jointly optimizes content selection and surface realization.	
<i>Oracle</i>	Konstas and Lapata (2012) present an end-to-end system that performs content selection and surface realization jointly.	
<i>Keyword</i>	Konstas and Lapata (2012) present an end-to-end generation model that performs content selection and surface realization jointly.	
<i>Context</i>	Konstas and Lapata (2012) proposed an unsupervised Concept-to-Text model that jointly optimizes content selection and surface realization. They represent their grammar compactly as a weighted hypergraph and perform decoding using a dynamic programming algorithm.	
<i>Gold</i>	Konstas and Lapata (2012) described an end-to-end generation model which jointly models content selection and surface realization.	
<b>Cited text spans</b>		
<i>Section</i>	<i>Sentence</i>	<i>Strategy</i>
Title	Unsupervised Concept-to-text Generation with Hypergraphs	c
Abstract	Concept-to-text generation refers to the task of automatically producing textual output from non-linguistic input.	ac
Abstract	We present a joint model that captures content selection ("what to say") and surface realization ("how to say") in an unsupervised domain-independent fashion.	aokc
Abstract	Rather than breaking up the generation process into a sequence of local decisions, we define a probabilistic context-free grammar that globally describes the inherent structure of the input (a corpus of database records and text describing some of them).	ac
Abstract	We represent our grammar compactly as a weighted hypergraph and recast generation as the task of finding the best derivation tree for a given input.	ac
Abstract	Experimental evaluation on several domains achieves competitive results with state-of-the-art systems that use domain specific constraints, explicit feature engineering or labeled data.	a
Introduction	A typical concept-to-text generation system implements a pipeline architecture consisting of three core stages, namely text planning (determining the content and structure of the target text), sentence planning (determining the structure and lexical content of individual sentences), and surface realization (rendering the specification chosen by the sentence planner into a surface string).	c
Introduction	Besides concentrating on isolated components, a few approaches have emerged that tackle concept-to-text generation end-to-end.	k
Introduction	Due to the complexity of the task, most models simplify the generation process, e.g., by creating output that consists of a few sentences, thus obviating the need for document planning, or by treating sentence planning and surface realization as one component.	c
Introduction	In this paper we describe an end-to-end generation model that performs content selection and surface realization jointly.	okc
Introduction	In order to ensure that our generation output is fluent, we intersect our grammar with a language model and perform decoding using a dynamic programming algorithm (Huang and Chiang, 2007).	c
Related Work	Our work is situated within the broader class of data-driven approaches to content selection and surface realization.	ok
Related Work	Given a corpus of database records and texts describing some of them, they learn a content selection model that simultaneously optimizes local label assignments and their pairwise relations.	c
Related Work	A few approaches have emerged more recently that combine content selection and surface realization.	o
Related Work	In contrast, Angeli et al. (2010) propose a unified content selection and surface realization model which also operates over the alignment output produced by Liang et al. (2009).	ok
Related Work	Similar to Angeli et al. (2010), we also present an end-to-end system that performs content selection and surface realization.	ok
Problem Formulation	Our goal then is to reduce the tasks of content selection and surface realization into a common probabilistic parsing problem.	ok
Results	We first assessed the performance of our generator (k-BEST) on joint content selection and surface realization and obtained a BLEU score of 24.88.	ok
Results	We observe a substantial increase in performance compared to the joint content selection and surface realization setting.	ok
Conclusions	We have presented an end-to-end generation system that performs both content selection and surface realization.	ok

Table 10: Citation text generation example: Gong et al. (2019) citing Konstas and Lapata (2012). The citation context is shown at the top, followed by the generated citation texts using the abstract and the three CTS strategies as inputs; the gold citation text is also shown. For each CTS sentence, “strategy” refers to the input setting(s) that included it: **abstract**, **oracle CTS**, **keyword CTS**, and **context CTS**.

<b>Citation context</b>	
This information can be incorporated into an SRL system in several different ways.	
<b>Citation text</b>	
<i>Abstract</i>	Swayamdipta et al. (2018) introduced the Syntactic scaffold, an approach to incorporating syntactic information into semantic tasks. The scaffold is designed to avoid the expensive syntactic processing at runtime, only making use of a treebank during training, through a multitask objective, and achieves competitive performance on PropBank semantics, frame semantics, and coreference resolution.
<i>Oracle</i>	Swayamdipta et al. (2018) use syntactic information to predict spans, but it is not a guarantee that a span is a valid parse tree.
<b>Cited text spans</b>	
<i>Abstract</i>	We introduce the syntactic scaffold, an approach to incorporating syntactic information into semantic tasks. Syntactic scaffolds avoid expensive syntactic processing at runtime, only making use of a treebank during training, through a multitask objective. We improve over strong baselines on PropBank semantics, frame semantics, and coreference resolution, achieving competitive performance on all three tasks.
<i>Oracle</i>	Span Features: • width of the span in tokens (Das et al., 2014) • distance (in tokens) of the span from the target • position of the span with respect to the target (before, after, overlap) Each of these features is encoded as a one-hot embedding and then linearly transformed to yield a feature vector, $a_i: j$ . <i>Introduction:</i> Since the scaffold task is not an end in itself, we relax the syntactic parsing problem to a collection of independent span-level predictions, with no constraint that they form a valid parse tree. Latent variables.: The motivation behind joint learning of syntactic and semantic representations is that any one task is helpful in predicting the other (Lluís and Màrquez, 2008; Lluís et al., 2013; Henderson et al., 2013; Swayamdipta et al., 2016) . Semi-Markov CRF: In order to model the non-overlapping arguments of a given target, we use a semi-Markov conditional random field (semi-CRF; Sarawagi et al., 2004) . Coreference Resolution: Labels $s_1$ to $s_m$ indicate a coreference link between $s$ and one of the $m$ spans that precede it, and null indicates that $s$ does not link to anything, either because it is not a mention or it is in a singleton cluster. Related Work: Using syntax as T2 in a pipeline is perhaps the most common approach for semantic structure prediction (Toutanova et al., 2008; Yang and Mitchell, 2017; Wiseman et al., 2016) . Coreference Resolution: Since there is a complete overlap of input sentences between Dsc and Dpr as the coreference annotations are also from OntoNotes (Pradhan et al., 2012), we reuse the $v$ for the scaffold task. Results: Tan et al. (2018) employ a similar approach but use feed-forward networks with selfattention. Results: Prior competitive coreference resolution systems (Wiseman et al., 2016; Clark and Manning, 2016b,a) all incorporate syntactic information in a pipeline, using features and rules for mention proposals from predicted syntax. Results: We follow the official evaluation from the SemEval shared task for frame-semantic parsing (Baker et al., 2007) .

Table 11: Citation text generation example, Wang et al. (2019b) citing Swayamdipta et al. (2018), showing the difference in compression ratio between the *Abstract* and *Oracle CTS* settings that results in unexpected QuestEval recall scores. The *Oracle* generation is supported by the *italicized CTS* sentence but still receives a low QuestEval score.