OARelatedWork: A Large-Scale Dataset of Related Work Sections with Full-texts from Open Access Sources

Martin Docekal, Martin Fajcik, Pavel Smrz
Brno University of Technology
{idocekal, ifajcik, smrz}@fit.vut.cz

Abstract

This paper introduces OARelatedWork: a dataset for related work generation from open-access sources.[1] It is the first large-scale multi-document summarization dataset for related work generation, containing whole related work sections and full texts of cited papers. Its validation and test splits are constructed so that every source paper is available in full text, enabling controlled evaluation of full-text related work generation. The dataset includes 94 450 papers and 5 824 689 unique referenced papers from multiple domains. With OARelatedWork, we aim to shift the field from generating parts of related work sections from abstracts only to generating entire related work sections from all available content. We (i) benchmark a wide spectrum of models, highlighting that synthesizing massive full-text contexts remains challenge for tested cost-effective Large Language Models (LLMs): under our statement-level judge, GPT-4o-mini's evidence-grounded *True* rate drops from 92.9% with abstracts to 83.8% with full texts. We (ii) empirically analyze human writing behavior through a human evaluation over 40 papers and 408 factual statements, revealing that authors frequently introduce abstractive claims ungrounded in localized source texts; consequently, tested LLMs actually surpass human baselines in strict, evidence-grounded factuality. Finally, we (iii) conduct a fine-grained meta-evaluation, revealing that standard reference-based metrics are inadequate for evaluating such long-form structured outputs, and introduce a robust statement-level evaluation framework to address this gap.

[1]Dataset: https://huggingface.co/datasets/BUT-FIT/OARelatedWork
Meta-Evaluation Dataset: https://huggingface.co/datasets/BUT-FIT/OARelatedWorkMetaEval
Main Repository: https://github.com/KNOT-FIT-BUT/OAPapers.

1 Introduction

Related work generation extends multi-document summarization by not only summarizing referenced papers but also positioning a work within prior research through comparisons and contrasts. Automating this process holds immense practical value: it aids researchers by drafting comparative summaries, significantly reducing the time spent on literature exploration. Furthermore, it pushes NLP models beyond simple summarization to handle complex, discourse-level generation, serving as a highly challenging benchmark for long-context reasoning.

As shown in Figure 1, related work generation is demanding due to extremely long input contexts, as models must process the target paper together with all cited references, while datasets must also provide access to all required papers, many of which may be paywalled. Furthermore, even when they are available, it is hard to automatically process them, as they are usually in PDF format. Fortunately, there are existing tools for conversion to more machine-friendly formats. For example, GROBID (GRO, 2008–2023), which was specifically designed to process scientific papers and corresponds to the state-of-the-art in its field (Meuschke et al., 2023).

Although the pioneering work of Hoang and Kan (2010) relied on full-text references and a topic hierarchy tree for a limited dataset, nowadays only abstracts are usually used (Lu et al., 2020; Chen et al., 2021, 2022; Liu et al., 2023a; Nishimura et al., 2024). This shift came probably with the raise of abstractive summarization models for which long inputs are technically demanding, given the constraints of existing models. Furthermore, the task of related work generation is commonly limited (Lu et al., 2020; Chen et al., 2021) to the generation of separate paragraphs. This makes the task easier, but also, such a simplification makes it impossi-

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 50728–50752
October 24-29, 2026 ©2026 Association for Computational Linguistics

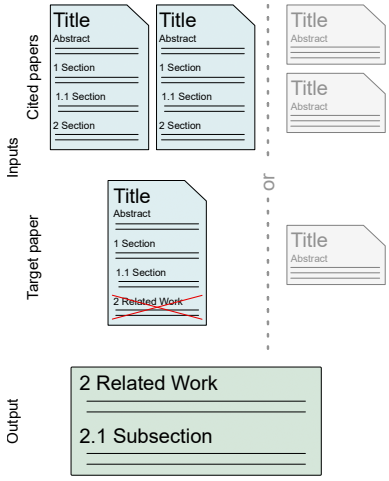


Figure 1: The task is to generate a whole related work section from cited papers and the rest of the target paper. We also try variants using abstracts instead of full-texts.

ble to assess whether a model is able to generate a well-structured related work section, as it only benchmarks models on a short sub-summary.

Generating an entire related work section is challenging not only for the generator but also for the evaluation. While automatic evaluation of summarization remains an open problem (Fabbri et al., 2021), the challenges are even more pronounced in the context of long-form summarization.

This work addresses the issues mentioned by:

- We introduce OARelatedWork (Section 2.2), providing automatically processed full texts of cited and target papers to benchmark the generation of entire related work sections. Through this dataset, we empirically analyze human-authored related work sections and compare them with generated ones.

- We evaluate diverse baselines (Section 6.2) and conduct a human evaluation (Section 7), revealing a tension between abstractive generation and factual faithfulness, while demonstrating that LLMs can surpass human baselines in strict evidence-grounding.

- We perform a fine-grained meta-evaluation (Section 7.1) of faithfulness metrics, proving the inadequacy of traditional reference-based metrics for this task and validating the use of LLM-based judges to reliably evaluate structured, long-form generation.

2 OARelatedWork Dataset

To obtain the final dataset, we first created a corpus of scientific articles. To establish our corpus, we used two sources, the *CORE* (Knoth et al., 2023; Knoth and Zdráhal, 2012) and *Semantic Scholar* (Lo et al., 2020; Kinney et al., 2023) corpora.

The developers of *CORE* supplied us with papers that had been converted into XML/TEI format using GROBID. There are 5.4 million documents up to the year 2020, each containing parsed full-text, bibliography, and other metadata such as title and authors. In contrast to Semantic Scholar data, these documents do not have an assigned year, field of study, or DOI. We used Microsoft Academic Graph (MAG) (Sinha et al., 2015) to obtain these fields. Also, the bibliography was not linked. There was no known ID of the paper mentioned, so we had to perform bibliography linking (more in Section 2.1).

Regarding the *Semantic Scholar* corpus, we used both the S2ORC (Lo et al., 2020) dataset and the papers available from the Semantic Scholar API (Kinney et al., 2023). In total, it consisted of 129.2M[2] documents up to 2023. The size is much larger than that of CORE because it also contains abstract-only papers. About 90% of papers are abstract-only.

We merged CORE and both Semantic Scholar corpora together[3], applied document structure and citations transformations (see Appendix A.2 and Appendix A.4), deduplication[4], and content filtering (Appendix A.5), and obtained an *OAPapers* collection of size 124.9M papers.

To represent each document, we use the JSON format, which contains a tree representation of the content of a document. There are multiple hierarchy levels for sections, subsections, paragraphs, and sentences. We believe that this tree representation might be beneficial for future use cases, as it allows you to easily select parts of a document or tag content parts at the input of a model.[5]

2.1 Bibliography Linking

We perform bibliography linking for two reasons. First, we used a data source containing documents without identified bibliography (CORE). The sec-

[2]This is the sum of 101.9 (API) and 27.3 (S2ORC). Thus, it is a number with duplicates.

[3]Preferring CORE version when there is a collision.

[4]We utilized searcher described in Appendix A.1.

[5]We also created CLI software, to better navigate in our data. It is available here https://github.com/KNOT-FIT-BUT/OAPapersViewer.

ond reason is to expand existing links. For this task, we used (i) *MAG*, (ii) the *Semantic Scholar citation graph*, and (iii) our *own searcher*.

We consider *Semantic Scholar* graph as our main source; however, during processing, we discovered that we can obtain additional links using also *MAG*. Finally, we used our *own searcher* implementation even though its contribution was marginal. The number of identified bibliographic entries increased only by 0.6 %, which is about 11.60 million new links. To evaluate our search, we decided to perform a manual evaluation of the results and found that the search was correct in 47 of 50 cases. See Appendix A.1 for more details.

2.2 Related Work Dataset

To create a suitable dataset for the generation of related work sections, we search in our corpus for documents containing related work-like section. This means that we are not looking just for sections with the *"Related Work"* headline, but also, e.g., *"Background"*, *"Related Literature"*, and *"Literature Review"*. The *"Background"* is only considered valid when there is another section named *"Introduction"* to avoid confusion.

In this way, we were able to find more than half a million documents with related work-like sections. However, we decided to filter them further to increase the quality of our final set. Our pipeline filters out each related work that (i) contains fewer than three sentences, (ii) cites fewer than two documents, or (iii) contains *citation group* without a linked document.

We define a *citation group* as a sequence of subsequent citation spans (e.g., *[1][4][6]*). The condition to groups is used because having full coverage of cited papers is unrealistic in an open-access setup. We believe that having at least one representative of each group is a good approximation, as it is often the case that citations in a group are cited for similar reasons.

We also make sure that every target document, for which the related work is generated, and all cited documents have an abstract.

In this way, we obtain 94 450 papers with a related work section, whose bibliographies together contain 5 824 689 unique cited papers (Table 1 reports the average number of source documents per related work section).

Of all of these unique cited papers, 82% are abstract-only papers. As we want to provide a dataset that is suitable for evaluation of models on full-texts, we decided to make sure that *every related work section in validation and test set splits has all source documents with full-text*. To achieve that, we found all documents satisfying this condition and used 50% of them as our test set, 30% of them as our validation set, and we assigned the rest to the train set.

Dataset	Samples (Tr/Va/Te)	Tgt. Words	Cited Words	Docs	Para.
S2ORC (Chen et al., 2021)	126k / 5k / 5k	148.0	1.1k	5.0	1.0
Delve (Chen et al., 2021)	73k / 3k / 3k	181.0	0.6k	3.7	1.0
Multi-XSci. (Lu et al., 2020)	30k / 5k / 5k	116.4	0.8k	4.4	1.0
STRoGeNS (Nishimura et al., 2024)	86k / - / -	514.3	3.0k	16.6	4.2
OARelatedWork (Ours)					
Train	91k	530.3	26.2k	11.4	4.5
Validation	1.1k	338.9	36.5k	6.5	3.2
Test	1.9k	336.5	36.6k	6.5	3.2

Table 1: Comparison of our dataset with existing related-work generation datasets.

However, as can be seen in Table 1, this way of splitting caused the train set to contain longer targets, as it is more probable to cover all citations in shorter text, because shorter related works have fewer citations on average. Nevertheless, this does not present a fundamental problem given our task definitions (see Section 3).

Domain Shift Our dataset exhibits a heavy shift toward computer science compared to the original corpus distribution (Appendix F), likely because Computer Science papers explicitly section "Related Work," whereas medicine and psychology integrate it into the introduction.

3 Task Definition

A system is provided with (1) a combination X of content types, including abstracts and full texts of the target paper (excluding the related work section) and its cited papers, (2) a target length L, and (3) information on whether the output should follow a structured format with subsections or a plain unstructured form.

Using these inputs, the task is to generate related work $Y = (y_1, y_2, ..., y_L)$, where y_i is the i-th token of the related work section with the length L. Target length and subsection information are provided because they are subjective decisions of an author rather than a feature of the input text.

To identify the contribution of using different inputs, we experiment with multiple input type com-

binations (shown later in Test set results Table 2).

4 Evaluation

We use several metrics to evaluate the generated text. To enable comparison with previous work, we use the ROUGE-1-F1, ROUGE-2-F1, and ROUGE-L-F1[6] metrics (Lin, 2004), and we also use BERTScore (BS) (Zhang et al., 2020).

We conducted human evaluation for selected configurations and use it for both evaluating models and meta-evaluating LLM judges (Section 7). From tested models we selected the quantized Gemma 4 31B (Team et al., 2026) without reasoning as our judge. This judge extracts statements and assigns them to one of four categories: *True*, *True but wrong citation* (hereafter referred to as *Wrong Cit.*), *Unverifiable*, or *False*. Further details are provided in Section 7.1.

Before applying an evaluation metric[7], we convert all citation and reference spans[8] to form `<cite>` and `<ref>` from their original form `<cite> ID <sep> TITLE <sep> FIRST_AUTHOR <sep> ... </cite>` and similar for references.

Citation Metric. We additionally compute F_1 between the sets of cited papers in the reference and the generation (full definition in Appendix B).

5 Related Work

Our work aligns with the survey by Li and Ouyang (2024). We benchmark their recommended baselines alongside modern LLMs, preserve complex multi-paper citations (Appendix A.4), and explore Retrieval-Augmented Generation (RAG) performance bounds via our greedy oracle experiments (defined in Section 6.1). Highlighting the field's shift toward full-text synthesis, Liu et al. (2025) recently adopted the preprint of our OARelatedWork dataset to benchmark their novel multi-agent generation framework. Other recent works also explore full-text inputs, but at small scale: Martin-Boyle et al. (2024) use GPT-4 to generate whole sections on 10 papers, and Li and Ouyang (2025) apply feature-based prompting on 27 papers. OARelatedWork fills the resulting need for a large-scale full-text whole-section benchmark.

5.1 Datasets

The very first work (Hoang and Kan, 2010) introduced related work generation with a dataset of 20 papers containing summary references, full-texts of cited articles, and topic trees, targeting whole related work sections. However, because abstractive approaches struggle with long inputs and outputs, subsequent datasets restricted inputs to abstracts and outputs to single paragraphs (Lu et al., 2020; Chen et al., 2021); a comparison with our data is available in Table 1.

A parallel line of work targets *citation text generation*, producing a sentence (or a few) citing a given reference from inputs such as the title and abstract of the cited paper (AbuRa'ed et al., 2020) or the citation context together with the cited abstract (Xing et al., 2020). Funkquist et al. (2023) released *CiteBench* to unify these datasets. Li et al. (2024) move beyond abstracts by retrieving cited text spans from the full text.

Operating at a larger scale, SurveyGen (Bao et al., 2025) targets the generation of complete survey papers, but, unlike our work, still lacks the full texts of cited references.

Months after the pre-print version of our dataset was publicly released[9], Nishimura et al. (2024) presented the *STRoGeNS* dataset of structured summaries that, like ours, targets whole related work sections rather than single paragraphs. However, it uses only abstracts, leaving our test set with more than 12× more input data (Table 1). Nevertheless, *STRoGeNS* has more cited documents on average. We believe that generating an entire related work section requires going beyond abstracts, when possible, and also providing the rest of the target paper to enable contrasting paragraphs (such as this one).

5.2 Evaluation

Related work generation is a summarization task (Hoang and Kan, 2010) and inherits its evaluation challenges, including low inter-annotator agreement, especially for non-experts (Fabbri et al., 2021; Gillick and Liu, 2010). Lexical-overlap metrics such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and METEOR (Lavie and Agarwal, 2007) are widely used but fail to match semantically equivalent texts with different surface forms. Model-based metrics like BERTScore (Zhang et al., 2020) and BARTScore (Yuan et al., 2021) miti-

[6] We use the summary-level variant.

[7] This does not hold for citation metrics and LLM judges.

[8] E.g., references to figures and tables.

[9] Our dataset was released in May 2024, while Nishimura et al. (2024) was in August 2024.

gate this but are limited by short context windows (e.g., 512 tokens for BERT), which is problematic for our long-form task. More recently, Liu et al. (2023b) proposed *G-Eval*, which prompts a large language model (GPT-4) with the task description, intermediate instructions, and a token-probability-based scoring function. Another worth mentioning is Prometheus 2 (Kim et al., 2024a), an open-weight LLM dedicated to evaluation.

Claim-level faithfulness evaluation. A complementary stream decomposes generations into atomic claims and verifies each one, starting with FActScore (Min et al., 2023) for biographies and extended to summarization by FENICE (Scirè et al., 2024), which combines claim extraction with NLI alignment, and MiniCheck (Tang et al., 2024), which trains efficient grounded fact-checkers. For long inputs, LongDocFACTScore (Bishop et al., 2024) and FaStFact (Wan et al., 2025) address the scalability of claim-based pipelines, while Zhong and Litman (2025) show that discourse-aware decomposition further improves the detection of factual inconsistency. FABLES (Kim et al., 2024b) performs claim-level human annotation of LLM-generated book-length summaries and finds that automatic raters correlate poorly with humans on long-form faithfulness. We adopt this paradigm in the related-work generation setting: our LLM judge extracts statements and assigns them to four citation-aware categories (*True*, *Wrong Cit.*, *Unverifiable*, *False*), and we meta-evaluate it against fine-grained human annotations (Section 7.1).

6 Experiments

We investigate several baselines on our dataset and ablate the impact of different types of information-rich inputs. These cover naive and oracle approaches, traditional systems, and transformer-based models.

6.1 Models

We evaluate three groups of models: extractive baselines and oracles, a fine-tuned multi-document summarization model, and zero-shot LLM baselines. Full implementation details, hyperparameters, and prompts are provided in the Appendix C, E, and P.

Extractive Baselines and Oracles We include standard extractive methods to establish baseline performance and estimate the upper bound of purely extractive summarization. LEAD selects the first sentence from each input abstract and appends the corresponding citation span. TEXTRANK (Mihalcea and Tarau, 2004) ranks sentences using lexical-overlap-based PageRank and is applied separately to each input document; the resulting sub-summaries are concatenated and assigned citation spans.

We also use a GREEDY ORACLE (GO), which iteratively selects sentences that maximize ROUGE-2 F1 against the reference summary (Nallapati et al., 2017). We evaluate GO both as a standalone extractive summarizer, where sentences are selected globally from all input documents, and as a retrieval component for neural models, where it produces one extractive sub-summary per document before concatenation.

PRIMERA PRIMERA (Xiao et al., 2022) is an LED-based (Beltagy et al., 2020) multi-document summarization model with 447M parameters, pre-trained using Gap Sentence Generation and Entity Pyramid Masking. We fine-tune PRIMERA on our task and report the average over three runs. Input length statistics and fine-tuning details are provided in Appendices D and C.

Zero-shot Models For zero-shot baselines, we evaluate GPT-4o mini (OpenAI et al., 2024) and Llama 3.2 3B (Grattafiori et al., 2024)[10]. As with PRIMERA, documents are represented using a Markdown-like format. The details are deffered to Appendix P.

6.2 Results

We overview the impact of different input types across models in Table 2.

Citation Coverage of GO: We omit citation concatenation for Greedy Oracle (GO) outputs to isolate how oracle sentences naturally cover cited documents. Using the full target yields higher citation scores, indicating that cited documents are frequently referenced in similar contexts outside the related work section (e.g., introductions). Notably, 16 % of citations are covered by salient sentences extracted solely from the target papers (assuming abstracts contain no citations).

Citation Formatting in Abstractive Models: Advanced LLMs (e.g., GPT-4o Mini) and trained models exhibit high citation scores, successfully

[10] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

Model	Input				citations	R1	R2	RL	BS	True	Wrong Cit.	Unverifiable	False
	References		Target Paper										
	Abs.	Cont.	Abs.	Cont.									
LEAD	✓				**99**	26	4	20	5	$99.1^{99.3}_{98.8}$	$0.2^{0.4}_{0.1}$	$0.6^{0.9}_{0.5}$	$0.0^{0.0}_{0.0}$
Greedy Oracle	✓				0	38	12	34	13	$98.1^{98.5}_{97.6}$	$0.1^{0.2}_{0.0}$	$1.1^{1.4}_{0.8}$	$0.6^{0.9}_{0.4}$
	✓	✓	✓	✓	15	**53**	**26**	**49**	**25**	$96.3^{96.7}_{95.7}$	$0.7^{0.9}_{0.5}$	$2.3^{2.7}_{1.9}$	$0.3^{0.5}_{0.2}$
	✓	✓	✓		7	49	23	46	21	$96.0^{96.5}_{95.3}$	$0.4^{0.6}_{0.3}$	$2.4^{2.8}_{2.0}$	$0.4^{0.5}_{0.3}$
	✓	✓			8	47	22	44	20	$94.9^{95.5}_{94.2}$	$0.3^{0.4}_{0.2}$	$3.2^{3.7}_{2.8}$	$0.4^{0.6}_{0.3}$
	✓		✓	✓	16	47	21	43	21	$97.6^{98.0}_{97.1}$	$0.4^{0.7}_{0.3}$	$1.2^{1.5}_{1.0}$	$0.5^{0.7}_{0.3}$
TextRank	✓				**99**	31	6	29	9	$99.4^{99.6}_{99.1}$	$0.2^{0.5}_{0.1}$	$0.3^{0.5}_{0.2}$	$0.0^{0.1}_{0.0}$
	✓	✓	✓	✓	65	31	5	27	7	$96.7^{97.1}_{96.1}$	$1.3^{1.7}_{1.1}$	$1.0^{1.4}_{0.7}$	$0.3^{0.4}_{0.2}$
	✓	✓	✓		67	34	6	30	9	$97.5^{97.8}_{97.0}$	$1.0^{1.3}_{0.8}$	$0.6^{0.8}_{0.4}$	$0.3^{0.6}_{0.2}$
	✓	✓			68	29	4	25	6	$95.1^{95.8}_{94.4}$	$1.8^{2.2}_{1.5}$	$1.3^{1.7}_{1.0}$	$0.2^{0.3}_{0.1}$
	✓		✓	✓	89	33	7	30	11	$99.5^{99.6}_{99.3}$	$0.1^{0.2}_{0.1}$	$0.3^{0.4}_{0.2}$	$0.0^{0.1}_{0.0}$
PRIMERA	✓		✓		96	35	8	32	13	$69.2^{70.1}_{68.4}$	$4.5^{4.8}_{4.2}$	$14.4^{15.0}_{13.8}$	$10.2^{10.6}_{9.8}$
	GO	GO	✓	GO	93	42	15	39	18	$65.1^{65.8}_{64.4}$	$7.1^{7.5}_{6.7}$	$12.3^{12.7}_{11.8}$	$13.1^{13.6}_{12.6}$
	GO	GO	✓		94	41	13	37	17	$63.4^{64.2}_{62.7}$	$6.0^{6.3}_{5.7}$	$14.3^{14.8}_{13.8}$	$13.6^{14.1}_{13.2}$
	✓		✓	GO	93	40	12	37	17	$69.1^{69.8}_{68.3}$	$5.7^{6.0}_{5.3}$	$12.4^{12.9}_{12.0}$	$11.7^{12.1}_{11.2}$
Llama 3.2 3B	✓		✓		2	29	6	26	5	$83.5^{84.3}_{82.7}$	$4.8^{5.3}_{4.4}$	$6.8^{7.3}_{6.3}$	$2.8^{3.1}_{2.5}$
	✓	✓	✓	✓	9	23	4	21	-1	$66.9^{68.1}_{65.6}$	$3.7^{4.2}_{3.4}$	$23.4^{24.6}_{22.2}$	$3.9^{4.2}_{3.6}$
	GO	GO	✓	GO	14	29	7	27	4	$71.0^{72.0}_{70.0}$	$5.8^{6.2}_{5.3}$	$14.2^{15.0}_{13.5}$	$6.2^{6.7}_{5.8}$
GPT-4o Mini	✓		✓		88	33	5	30	12	$92.9^{93.4}_{92.3}$	$2.8^{3.2}_{2.4}$	$2.7^{3.0}_{2.4}$	$0.8^{0.9}_{0.6}$
	✓	✓	✓	✓	73	32	5	29	11	$83.8^{84.6}_{82.9}$	$6.0^{6.5}_{5.5}$	$6.0^{6.5}_{5.6}$	$1.2^{1.4}_{1.0}$
	GO	GO	✓	GO	78	35	6	31	14	$81.4^{82.2}_{80.5}$	$8.6^{9.3}_{8.0}$	$5.8^{6.2}_{5.3}$	$2.0^{2.3}_{1.8}$
Human (Gold)	-	-	-	-	-	-	-	-	-	$73.5^{74.6}_{72.4}$	$1.6^{1.9}_{1.4}$	$18.5^{19.5}_{17.6}$	$2.3^{2.6}_{2.1}$

Table 2: Test set results on multiple input variants. Inputs with Greedy Oracle (GO) are texts pre-extracted using Greedy Oracle extractor. Factuality metrics (True, Wrong Cit., Unverifiable, False) include 95% bootstrap confidence intervals formatted as $Mean^{Upper}_{Lower}$. Missing percentages reflect judge output errors.

synthesizing information across multiple papers rather than collapsing to a single source. Conversely, we observed that smaller models like Llama 3.2 often struggle to strictly adhere to the requested citation format despite explicit prompt instructions.

Faithfulness & The Extractive Illusion: Our meta-evaluated LLM judge suggests a tension between extractive evidence preservation and abstractive generation. Extractive baselines obtain very high *True* scores (95–99%), likely because their outputs closely reuse source text; however, these scores should not be interpreted as evidence of better coherence or synthesis. By contrast, the human-written gold summaries receive a lower strict evidence-grounded *True* score (73.5%) and a higher *Unverifiable* rate (18.5%) than the extractive baselines. This does not imply that the human summaries are mostly false: their *False* rate remains low (2.3%). In Appendix K, our manual analysis of the unverifiable human-written statements identifies three causes: missing citations, automated parsing errors, and claims not grounded in the available source material. Among the 23 unverifiable statements analyzed, 7 were due to missing citations, 6 to parsing errors, and 10 to ungrounded claims. Among abstractive systems, GPT-4o Mini achieves the highest evidence-grounded factuality, while PRIMERA produces more than 10% *False* statements in several settings.

Influence of Available Content & Citation Conflation: Expanding input context paradoxically degrades factual accuracy for abstractive LLMs. Providing full texts instead of just abstracts caused GPT-4o Mini's *True* score to drop from 92.9% to 83.8%, accompanied by spikes in *Wrong Citation* (correct claims misattributed to the wrong source) and *Unverifiable* rates. Applying GO as a dense Oracle RAG step mitigated noise for the smaller Llama 3.2 (improving factuality) but further degraded performance for GPT-4o Mini.

7 Human Evaluation

We conducted a human evaluation of PRIMERA, GPT-4o-mini, and human-written reference sections across 40 randomly selected papers, yielding

120 related work sections.[11] Both models generated text using the greedy oracle setup with all available references and target papers.

Evaluating generated related work sections is a high-cognitive-load task, requiring annotators to read and synthesize context from multiple source papers. Given the significant human labor costs involved, scaling the sample size beyond 40 target papers (120 sections) was practically constrained. To account for this and rigorously support our claims, we prominently report confidence intervals for evaluation metrics, demonstrating the statistical reliability of our findings.

To evaluate factuality, we extracted factual statements from each section using GPT-5 (Prompt 1, Appendix P), which extracted verbatim spans and initially classified them as *True* or *False*. We mapped these statements back to the source text using fuzzy matching (Dice–Sørensen coefficient, threshold 0.8), successfully recovering 84% of statements (42% as exact matches).

To manage annotation workload, we randomly sampled up to three statements per section. To improve discriminability, we additionally constructed an adversarial subset using GPT-5-mini and GPT-OSS-20B. Statements with conflicting factuality labels across the extraction models were flagged as adversarial candidates. We sampled up to three adversarial statements per section and added them to the pool, yielding 408 statements for evaluation. Notably, human-written statements comprised only 16.4% of this adversarial subset.

We employed a pairwise evaluation format (Figure 7, Appendix G). First, at the *statement level*, annotators classified statements into four categories: *True* (supported by sources), *True but wrong citation* (factually correct but attributed to the wrong source), *False* (contradicted by sources), or *Unverifiable* (ambiguous, nonsensical, or lacking evidence).

Next, at the *document level*, annotators indicated their preference (or a tie) between the two sections across four dimensions: (i) **Relevance**: which section more directly and thoroughly relates to the target paper's topic, contributions, and context; (ii) **Faithfulness**: which section is more factually accurate and avoids incorrect claims; (iii) **Language**: which section is clearer, more grammatically correct, and easier to read; and (iv) **Overall Ranking**: the section preferred overall.

Detailed inter-annotator agreement scores are provided in Appendix I. Notably, the seemingly low agreement for *Language* ($\kappa = 0.1707$) stems from the *kappa paradox*: annotators overwhelmingly selected "no preference," making strict disagreements exceedingly rare. Furthermore, agreement dropped notably on the adversarial subset ($\kappa = 0.4368$) compared to randomly selected statements ($\kappa = 0.6556$), confirming these edge cases are difficult for humans to verify.

We observe higher agreement for fine-grained statement-level faithfulness ($\kappa = 0.6169$) than for coarse document-level Faithfulness ($\kappa = 0.4301$). This aligns with LongEval (Krishna et al., 2023), which found that evaluating faithfulness at a finer granularity yields more reliable judgments for long texts, corroborating our decision to fact-check localized statements.

Table 3 presents the pairwise results for document-level annotators. GPT-4o-mini overwhelmingly dominates, significantly outperforming both PRIMERA and human authors. This aligns with recent findings on LLM superiority in summarization (Pu et al., 2023). Notably, GPT-4o-mini's dominance holds across all attributes except *Faithfulness*, where it statistically tied with PRIMERA. In Appendix J, we further explore *Faithfulness* at the statement level.

To understand these preferences, we analyzed the linguistic properties of the generated texts. Flesch Reading Ease scores indicate that GPT-4o-mini generates exceptionally dense, complex text (-1.05), compared to humans (23.58) and PRIMERA (32.27). This rich vocabulary likely triggers a known annotator bias where judges inherently favor articulate, authoritative-sounding text (Gudibande et al., 2023). Additionally, we assessed extractiveness via ROUGE-2 overlap with the source texts. PRIMERA exhibits a strong extractive tendency (0.9157), whereas human writers (0.6502) and GPT-4o-mini (0.5217) are significantly more abstractive.

Our coarse evaluation reveals that PRIMERA is statistically indistinguishable from human authors across all four attributes. We hypothesize this parity is heavily driven by the Greedy Oracle framework providing an idealized, highly relevant context. While further ablation is required, this suggests that pairing a weaker sequence-to-sequence model with a strong retrieval mechanism may compensate for its generative limitations. Nevertheless,

[11] Annotator recruitment and compensation are detailed in Appendix N.

Attribute	Model Pair (A vs. B)	TWR (A)	95% CI
Preference	**GPT-4o-mini** vs. PRIMERA	**0.825**	[0.725, 0.913]
	Human vs. PRIMERA	**0.456**	**[0.338, 0.581]**
	GPT-4o-mini vs. Human	**0.800**	[0.700, 0.894]
Relevance	**GPT-4o-mini** vs. PRIMERA	**0.756**	[0.656, 0.844]
	Human vs. PRIMERA	**0.488**	**[0.381, 0.600]**
	GPT-4o-mini vs. Human	**0.788**	[0.694, 0.869]
Faithfulness	**GPT-4o-mini vs. PRIMERA**	**0.600**	**[0.481, 0.713]**
	Human vs. PRIMERA	**0.475**	**[0.356, 0.588]**
	GPT-4o-mini vs. Human	**0.700**	[0.600, 0.800]
Language	**GPT-4o-mini** vs. PRIMERA	**0.631**	[0.581, 0.681]
	Human vs. PRIMERA	**0.506**	**[0.438, 0.575]**
	GPT-4o-mini vs. Human	**0.625**	[0.575, 0.675]

Table 3: Pairwise human document level evaluation results based on True Win Rate (TWR; ties award 0.5 points). Winning models are highlighted in bold. When a tie occurs (the neutral baseline of 0.500 falls within the 95% Bootstrap Confidence Interval), the entire row is bolded.

Appendix J shows that PRIMERA falls behind under statement-level faithfulness evaluation.

To establish a reliable foundation for our LLM judge meta-evaluation, we performed a fine-grained, statement-level faithfulness evaluation. Because scanning large text volumes introduces high cognitive load, our evaluation process included a targeted meta-annotation round to resolve initial disagreements and correctly categorize unverifiable claims. The detailed methodology, analyzes of unverifiable statements, and full evaluation results (including the complete performance table) are deferred to Appendix J. Ultimately, this fine-grained analysis demonstrates that GPT-4o-mini beats human-written related work in factuality, a finding that aligns with Pu et al. (2023), who similarly observed high human hallucination rates.

7.1 Meta-Evaluation

We used the statement-level annotations to meta-evaluate candidate faithfulness metrics. We instructed LLM judges (Appendix H.1, Prompt 3) to extract and classify statements, and to provide evidence and rationale, so that each decision becomes interpretable. We calculate the final coarse faithfulness as proportion of true statements in all extracted statements in given related work. This coarse-level transformation also allowed us to meta-evaluate LLM metrics together with metrics like ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020). See Figure 2 for the results.

Compared to LongEval baselines (Krishna et al., 2023), non-LLM metrics performed poorly. Notably, the reference-based setup exhibited distribution shifts toward lower correlations, indicating

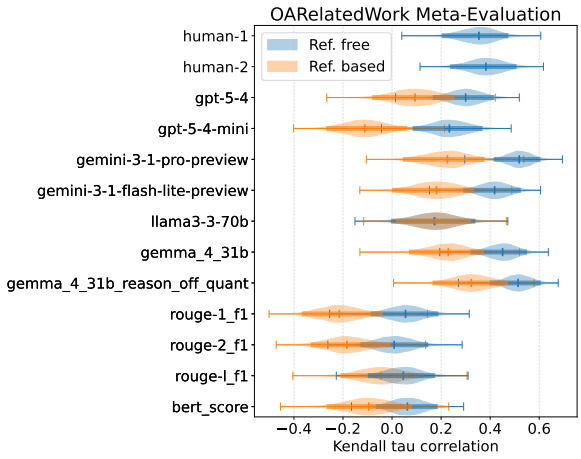


Figure 2: Violin plot of bootstrap distributions of Kendall's τ correlations of automatic metrics with human judgment on OARelatedWork meta-evaluation dataset for statement-level Faithfulness. Thick line is showing 95% confidence interval.

that while they might suffice for validating abstract generation (see Appendix L for more details), the reference-based metrics are inadequate for related work generation.

Interestingly, the Figure 2 shows that disabling reasoning for Gemma 4 did not degrade performance. We hypothesize this is because our output format inherently forces the model to provide evidence and a rationale, though confirming this requires further ablation. For the final evaluation, we utilized Gemma 4 without reasoning as a cost-effective alternative to Gemini 3.1 Pro.

We additionally report a meta-evaluation of F1 scores for pre-extracted statements in Appendix M.

7.1.1 Parametric Knowledge Experiment

To investigate reliance on internal parametric knowledge versus the strictness of context-bound evaluation, we conducted an ablation study restricting the judge model (Gemini 3.1 Pro) to verifying individual facts using only the greedy oracle inputs. Under these constraints, *Unverifiable* statements increased across all sources. For GPT-4o-mini, they rose from 6.80% to 17.69%, indicating that even with an idealized context, the LLM moderately relies on its pre-trained weights to hallucinate external facts. Due to its extractive nature, PRIMERA exhibited the smallest shift (4.29% to 7.86%).

Most revealing was the human baseline: when evaluated strictly against the narrow greedy oracle extracts, its *Unverifiable* rate increased from 19 % to 45 %. We conjecture, based on manual inspection, that human authors frequently integrate

information distributed across the document rather than relying on localized evidence spans. As a result, evaluation against strictly extractive context windows may systematically overestimate unverifiability for human-written summaries.

We hypothesize that this "Unverifiable delta" could be used to empirically calibrate optimal chunk sizes and refine retrieval strategies of RAG pipelines.

8 Conclusion and Future Work

Our work shifts related-work generation toward a realistic, full-section paradigm by leveraging comprehensive full-text contexts. Our human evaluation provides insights into academic writing, revealing that human authors are highly abstractive and frequently introduce claims ungrounded in localized source texts. Consequently, advanced LLMs can actually outperform human writers in strict factual faithfulness. Furthermore, our meta-evaluation clearly demonstrates that standard reference-based metrics are insufficient for evaluating complex, long-form related work generation. By providing structured data, fine-grained factual annotations, and a specialized statement-level evaluation framework, we establish a robust foundation for future research.

Limitations and Ethical Considerations

Data scope. The dataset is limited to open-access papers and exhibits a heavy shift toward computer science (Section 2.2, Appendix F). Our findings about generator and judge behavior are therefore best calibrated to CS-style related work and may not transfer cleanly to other disciplines. The dataset is text-only, which might be limiting when the essential information is, e.g., in a plot.

Document processing. Automatic parsing of PDFs into structured text is imperfect, and processing errors may miss or alter information. Our manual analysis (Appendix K) finds parsing-related errors in only ~5% of human-written statements, indicating a reliable but not flawless pipeline.

Potential data contamination. The papers in OARelatedWork are drawn from open-access corpora. It is therefore plausible that used LLM models have already seen used content.

Ethical use. The baseline models are intended as research tools for professionals; their outputs should be reviewed critically.

Acknowledgement

This work was supported by the EU CHIST-ERA project SoFAIR – Making Software FAIR: Identifying and Extracting Software Mentions from Open Research Papers and Registering them with PIDs (the Technology Agency of the Czech Republic, Grant No. TH86010002, 2024-2025).

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254)

References

2008–2023. Grobid. https://github.com/kermitt2/grobid.

Ahmed Ghassan Tawfiq AbuRa'ed, Horacio Saggion, Alexander V. Shvets, and Àlex Bravo. 2020. Automatic related work section generation: experiments in scientific document abstracting. *Scientometrics*, 125:3159 – 3185.

Tong Bao, Mir Tafseer Nayeem, Davood Rafiei, and Chengzhi Zhang. 2025. SurveyGen: Quality-aware scientific survey generation with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2712–2736, Suzhou, China. Association for Computational Linguistics.

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Jennifer A. Bishop, Sophia Ananiadou, and Qianqian Xie. 2024. LongDocFACTScore: Evaluating the factuality of long document abstractive summarisation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10777–10789, Torino, Italia. ELRA and ICCL.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and 1 others. 2024. Symbolic discovery of optimization algorithms. *Advances in Neural Information Processing Systems*, 36.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Rui Yan, Xin Gao, and Xiangliang Zhang. 2022. Target-aware abstractive related work generation with contrastive learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 373–383, New York, NY, USA. Association for Computing Machinery.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang 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.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Martin Funkquist, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. 2023. CiteBench: A benchmark for scientific citation text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7337–7353, Singapore. Association for Computational Linguistics.

Dan Gillick and Yang Liu. 2010. Non-expert evaluation of summarization systems is risky. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151, Los Angeles. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *Preprint*, arXiv:2305.15717.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *International Conference on Computational Linguistics*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024a. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Yekyung Kim, Yapei Chang, Marzena Karpinska, Aparna Garimella, Varun Manjunatha, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024b. Fables: Evaluating faithfulness and content selection in book-length summarization. *Preprint*, arXiv:2404.01261.

Rodney Michael Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Rob Evans, Sergey Feldman, Joseph Gorney, David W. Graham, F.Q. Hu, and 29 others. 2023. The semantic scholar open data platform. *ArXiv*, abs/2301.10140.

Petr Knoth, Drahomira Herrmannova, Matteo Cancellieri, Lucas Anastasiou, Nancy Pontika, Samuel Pearce, Bikash Gyawali, and David Pride. 2023. Core: A global aggregation service for open access papers. *Scientific Data*, 10.

Petr Knoth and Zdenek Zdráhal. 2012. Core: Three access levels to underpin open access. *D Lib Mag.*, 18.

Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. LongEval: Guidelines for human evaluation of faithfulness in long-form summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1650–1669, Dubrovnik, Croatia. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Xiangci Li, Yi-Hui Lee, and Jessica Ouyang. 2024. Cited text spans for scientific citation text generation. In *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, pages 90–104, Bangkok, Thailand. Association for Computational Linguistics.

Xiangci Li and Jessica Ouyang. 2024. Related work and citation text generation: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13846–13864, Miami, Florida, USA. Association for Computational Linguistics.

Xiangci Li and Jessica Ouyang. 2025. Explaining relationships among research papers. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1080–1105, Abu Dhabi, UAE. Association for Computational Linguistics.

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.

Jiachang Liu, Qi Zhang, Chongyang Shi, Usman Naseem, Shoujin Wang, and Ivor Wai-Hung Tsang. 2023a. Causal intervention for abstractive related work generation. *ArXiv*, abs/2305.13685.

Xiaochuan Liu, Ruihua Song, Xiting Wang, and Xu Chen. 2025. Select, read, and write: A multi-agent framework of full-text-based related work generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7009–7028, Vienna, Austria. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

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.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-xscience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Conference on Empirical Methods in Natural Language Processing*.

Anna Martin-Boyle, Aahan Tyagi, Marti A. Hearst, and Dongyeop Kang. 2024. Shallow synthesis of knowledge in gpt-generated texts: A case study in automatic related work composition. *Preprint*, arXiv:2402.12255.

Norman Meuschke, Apurva Jagdale, Timo Spinde, Jelena Mitrović, and Bela Gipp. 2023. A benchmark of pdf information extraction tools using a multi-task and multi-domain evaluation framework for academic documents. *ArXiv*, abs/2303.09957.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

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.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: a recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Kazuya Nishimura, Kuniaki Saito, Tosho Hirasawa, and Yoshitaka Ushiku. 2024. Toward structured related work generation with novelty statements. In *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, pages 38–57, Bangkok, Thailand. Association for Computational Linguistics.

Ollama. 2023. Ollama: Run large language models locally. `https://ollama.com`. Accessed: 2025-05-19.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *Preprint*, arXiv:2309.09558.

Alessandro Scirè, Karim Ghonim, and Roberto Navigli. 2024. FENICE: Factuality evaluation of summarization based on natural language inference and claim extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14148–14161, Bangkok, Thailand. Association for Computational Linguistics.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 243–246, New York, NY, USA. Association for Computing Machinery.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, and 304 others. 2026. Gemma 4 technical report. *Preprint*, arXiv:2607.02770.

Yingjia Wan, Haochen Tan, Xiao Zhu, Xinyu Zhou, Zhiwei Li, Qingsong Lv, Changxuan Sun, Jiaqi Zeng, Yi Xu, Jianqiao Lu, Yinhong Liu, and Zhijiang Guo. 2025. FaStFact: Faster, stronger long-form factuality evaluations in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23814–23854, Suzhou, China. Association for Computational Linguistics.

Kilian Weinberger, Anirban Dasgupta, Josh Attenberg, John Langford, and Alex Smola. 2010. Feature hashing for large scale multitask learning. *Preprint*, arXiv:0902.2206.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. pages 38–45. Association for Computational Linguistics.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. 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.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

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*.

Yang Zhong and Diane Litman. 2025. Discourse-driven evaluation: Unveiling factual inconsistency in long document summarization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2050–2073, Albuquerque, New Mexico. Association for Computational Linguistics.

A Data Processing

A.1 Searcher for Bibliography Linking

Our search is done in multiple stages. The first stage is candidate retrieval. It consists of a string-match search on document titles using a SQLite database. We allow different additional postfixes for the matched title. Additionally, approximate matching is performed. It is looking for the k-nearest neighbors in the vector space of embeddings created using the feature locality-sensitive hashing (Weinberger et al., 2010). We used this method because of its low computational complexity.

The last step is the filtering stage. The filter works with the title, authors, and year of a record. The title and authors fields are mandatory. If the year field is known for both records that are supposed to be matched, there cannot be more than a two-year difference. Following Lo et al. (2020), we calculate the similarity score for a *title* as the harmonic mean of the Jaccard index and containment metric:

$$C = \frac{|N_1 \cap N_2|}{\min(|N_1|, |N_2|)}, \quad (1)$$

where N_1 and N_2 are word sets from matched strings, which are also used for the Jaccard index.

Only titles with similarity above the 0.75 threshold are considered as a match. When matching *authors*, we require that at least one pair of authors have a match. The match is again done using a similarity metric (using the same threshold), but now it is just the containment metric because we want to obtain high similarity even when there is just, e.g., a surname in one source. We observe that it is often the case that the names are expressed using initials in one source and the full form in the other. For that reason, we also create an initial version of each name and consider it when matching.

For similarity computation via Jaccard / containment metric and string-match search, we first normalize the title and authors (authors are not used for string-match search) by converting the string into its close ASCII form using Unidecode[12], removing non-word characters[13], converting to lower case, and removing repeating characters.

A.1.1 Estimation of the Error Effect of the Fallback Searcher

Our manual 50-sample evaluation of the fallback searcher yielded an accuracy of 94% (47/50 correct). As noted in Section 2.1, the fallback searcher contributed a marginal 0.6% of the total identified links (the remaining 99.4% were resolved via robust MAG and Semantic Scholar graphs). Under the conservative Clopper–Pearson exact method,

[12] https://pypi.org/project/Unidecode/

[13] The \W python regex placeholder.

the lower bound of the confidence interval for accuracy is 0.8345 (upper bound: 0.9875). Even under this worst-case statistical bound, the total potential error introduced to the overall dataset by the fallback searcher is strictly less than 0.1% ($0.006 \times 0.1655 \approx 0.099\%$).

A.2 Content Hierarchy

In the sources used, the documents were already parsed into sections and paragraphs. Nevertheless, we decided to add more levels to this shallow hierarchy, as we believe that it will increase the usability of our data. Ideally, we would like a document divided into sections, n-levels of subsections, paragraphs, and sentences. To segment sentences, we utilized the scispaCy model (Neumann et al., 2019).

The problem of parsing subsections is very difficult, as it is often the case that section headlines from existing resources are already not parsed correctly. We used a custom heuristic based on the numbering present in headlines. This heuristic falls back to a shallow hierarchy when parsing fails. More details are presented in Appendix A.3.

We manually analyzed our subsection parser on 100 papers. As we wanted only to evaluate the subsection parser, we only considered the provided headlines from the original corpora and not the true section headlines from the original PDF. We saw that for 20 papers, the algorithm failed and left the original shallow hierarchy. From the rest of the 80 papers, we considered 89 % parsed correctly. Our analysis also showed that 55% of these 80 papers did not use subsections.

A.3 Subsection Parser

To parse subsections, we use the numbering present in a headline. Our method is applied to the list of headlines, some of which might be subsections. It first tries to create an internal numbering by guessing section numbers for headlines with missing numbers by exploiting writing habits and format. See Figure 3 for an example of conversion. Then comes identifying the longest sparse[14] sequence of consecutive numberings (which we call the anchor). Then, it fits the rest of the headlines into a constructed hierarchy, as in the example in Figure 3 where the non-bold headlines are put in the *II.3 metrics* section. The algorithm leaves the original, shallow hierarchy if it fails. The method stays

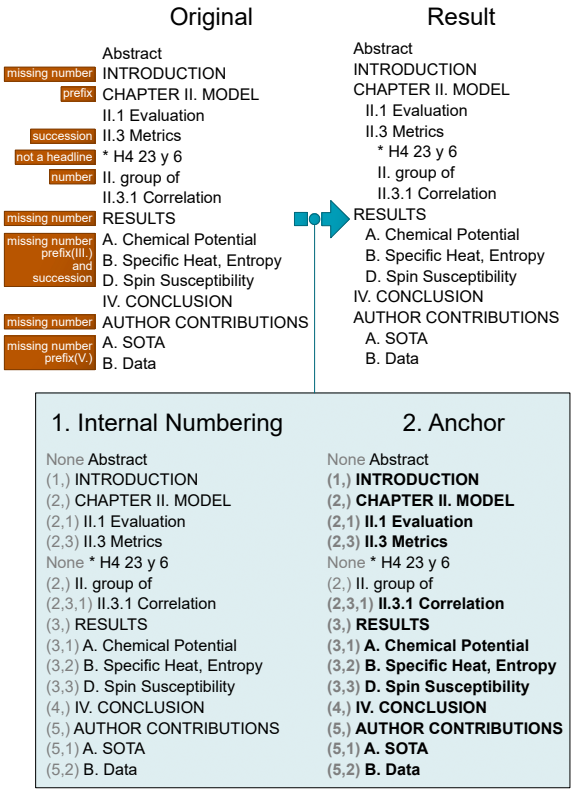


Figure 3: An example of hierarchy parsing showing two major steps: Internal Numbering and creation of Anchor. Orange boxes contain notes with obstacles for the parser. See that we are able to guess section numbers for headlines with missing numbers by exploiting writing habits and format. The introduction is usually the first section, and we can see that the capital letters format is used for all (known) top-level headlines. The headline * *H4 23 y 6* will be removed in the cleaning step.

with the original when the anchor consists of less than three elements or when the numbering is not matching levelwise. For example, assume that the previous subsection was 2.3.1 and that section 3.5 follows. It also considers the numeric format; thus e.g., for previous subsection 2.3.1, the following subsection 2.3.B would be considered problematic. Because an exhaustive description of the conversion would be beyond the scope of this paper, we direct the reader to investigate our code.

To perform sentence segmentation, we used the scispaCy model (Neumann et al., 2019). It was trained on papers from the biomedical domain, so it is more suitable for our case as standard (web) models usually have problems with this type of document. One of the problems with standard models is that the citation spans are misclassified as sentence boundaries.

[14] We allow at most 3 missing numbers in between.

A.4 Citation Spans

In our data, we leave the citation spans in their original form, except for the merged citations (e.g., [3-6]). We decided to expand these to each individual citation, as we are providing character offsets to identify the citation span. Otherwise, it would mean that two citation spans could be in the same position.

The original sources already provide parsed citation spans. However, we decided to also employ our own simple parser for Harvard-style citations. It is a regular expression searcher that checks whether the information in searched spans corresponds to at least a single bibliographic entry. The matching is done on author and year. The year must match exactly, and the matching of the author is done in the same way as described in Section 2.1. This step is done after bibliography identification. We found an additional 13.66 million citation spans, which is a 1.8% increase.

A.5 Document Content Cleaning

As we are processing documents that might be badly parsed, we decided to add a cleaning step to our data processing pipeline. We remove:[15]

- Sections with empty headline.
- Sections without text in their whole subhierarchy.
- Sections with headline with less than 30% of Latin characters.

B Citation Metric

As citations are an important component of related work sections, we evaluate them separately. Let R_c denote the set of cited documents in the reference (i.e., titles or identifiers of cited papers[16]), and let P_c denote the set of citations in the generated text. We measure citation quality using recall $c_r = |R_c \cap P_c|/|R_c|$ and precision $c_p = |R_c \cap P_c|/|P_c|$, and report their F_1 score.

C PRIMERA Fine-Tuning

To fine-tune, we use the Lion (Chen et al., 2024) optimizer with a linear scheduler with 4 000 warmup steps, learning rate 1e-05, no weight decay, batch size 16, and applied early stopping with patient 3. We use teacher forcing during training and beam search with beam size 4 during inference.

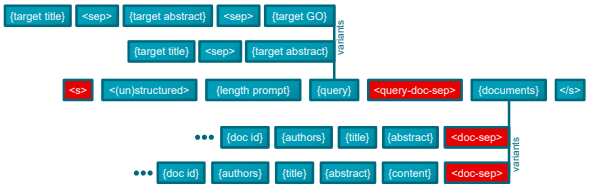


Figure 4: Illustration of input format for PRIMERA. The red boxes are global tokens. We put the special tokens `<unstructured>` or `<structured>` to identify whether we want to generate subsections. The `length prompt` is the number of tokens (in text form) of a reference target sequence. GO means content filtered by the Greedy Oracle model. In experiments, the GO is also used for filtering full-text of cited documents.

Figure 4 shows the format of input. It can be seen that we added one new global token `<query-doc-sep>` for dividing query and content of cited documents. Document content, also target content, is represented using Markdown-like text with a special format of citations and references (see Section 4). When content is not available, we put at least authors (max two) and title at the input. Each citation constitutes from id or unknown symbol, title of cited paper, and first author. This format allows models to draw the references from their memory.

Because we work with long inputs consisting of scientific papers (see Appendix D for detailed length analysis) and the original model supports only inputs of maximal length 4 096 tokens and output of 1 024 tokens, we increased the number of encoder positional embeddings to 8 192 (16 384 for GO experiments) and 2 048 for the decoder. As in Beltagy et al. (2020) we initialized these additional embedings by copying the already existing ones.

D Input Length Analysis

Figure 5 shows the length analysis of different inputs.

E Other Used Tools

This section describes libraries, frameworks, or tools used during the creation of this work that were not already mentioned.

We use the Hugging Face (Wolf et al., 2020) library for obtaining ROUGE. We also used Hugging Face implementation of PRIMERA model. The TextRank implementation is taken from the

[15] By sections we also mean subsections.

[16] We omit citations without resolvable bibliographic entries for clarity.

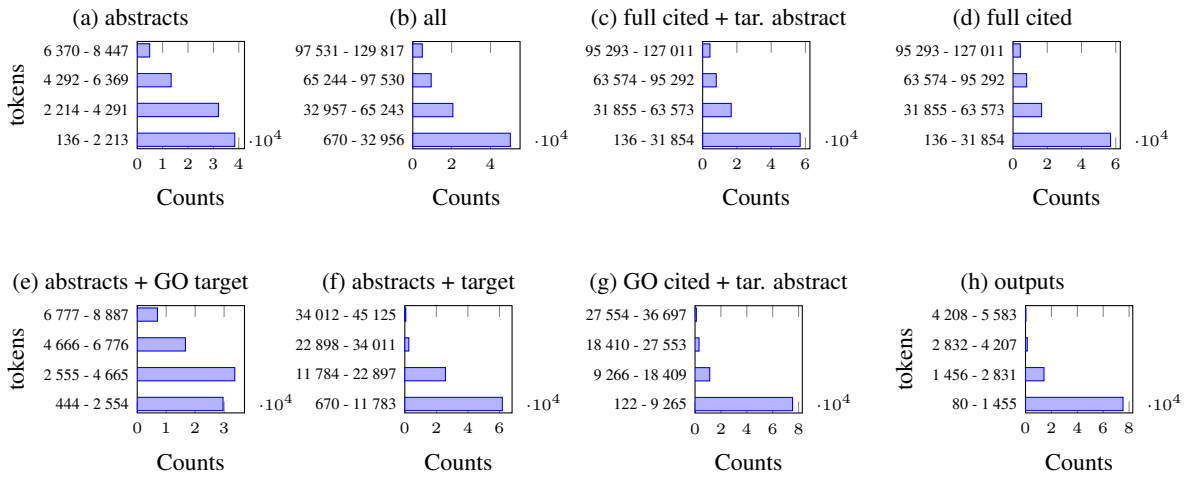


Figure 5: Histograms showing lengths of input types and lengths of outputs on train set. These numbers are obtained from PRIMERA tokenized inputs/outputs. GO for target paper is not created from abstract. GO summaries of cited documents are created for each document separately. We cut the top part of histograms to fit them on the page, thus the maximum is not there.

summa Python package (Barrios et al., 2016). We use Ollama (Ollama, 2023) for the Llama 3.2 3B. The GPT-4o Mini results were obtained using the official OpenAI API `https://openai.com/`.

F Domain shift

This section provides the detailed distribution of study fields across the original corpus and our resulting dataset splits. As shown in Figure 6, while the original un-filtered corpus covers diverse disciplines (such as medicine and biology), the structural requirement of having dedicated related work-like sections introduces a pronounced bias. Consequently, computer science heavily dominates the final training, validation, and test sets.

G Annotation Instructions

G.1 Objective

The goal of this task is to **evaluate the quality of Related Work sections** for research papers.

G.2 Materials Provided

- **Target Paper:** The main paper for which you are evaluating related work.
- **Two Versions of Related Work:** Competing drafts. You will select the version you consider better.
- **Cited Papers:** References to support citations in the related work sections.

G.3 Annotation Process

1. Familiarize Yourself Read the **abstract of the target paper** to understand its main contributions and scope.

2. Review Related Work Sections Carefully read **both versions** of the related work section.

3. Annotate Factual Statements

- There are **up to 6 factual statements** for each related work section.
- For each statement, choose:
  - **True** – The statement is supported by the target or cited papers.
  - **True, but wrong citation** – The statement is factually correct and supported by a different provided source, but not by the one it cites.
  - **False** – The statement contradicts the provided sources.
  - **Unverifiable** – The statement cannot be verified from the sources, is ambiguous, or nonsensical.
- **Important:**
  - Use information **only** from the target paper and cited papers.
  - If a given statement span is not correct, create a new one and refer this new span to the original one (you can use `Alt+R`).

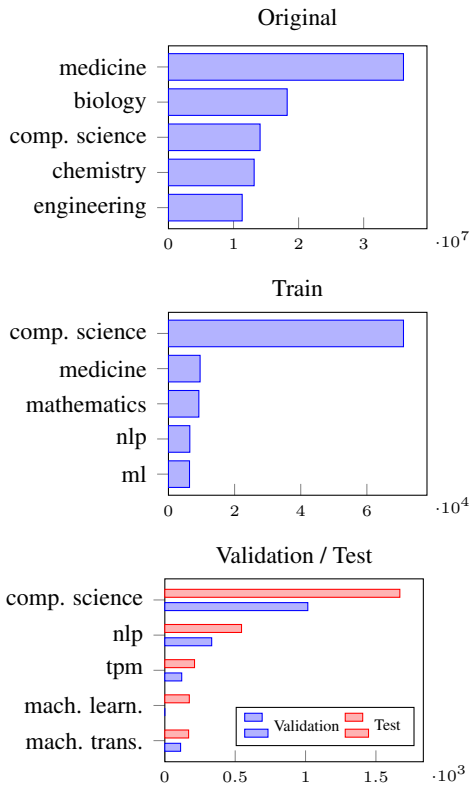


Figure 6: Field of study domain shift between all papers in used corpus and dataset splits. As a single paper may have multiple fields of study, the counts do not add up to the total number of papers. The *tpm* is *task (project management)*.

  - Use the metadata field (`Alt+M`) if you want to add a comment about your decision.
  - If you do not understand, select "Unverifiable" and write "I do not understand" in the metadata field (`Alt+M`).

4. Select Preferred Related Work After reviewing all factual statements, decide which of the two provided related work sections you prefer for each evaluation aspect.

Evaluation Aspects:

- **Relevance** – Choose the related work section that more directly and thoroughly relates to the target paper's topic, contributions, and context.
- **Faithfulness** – Choose the section that is more factually accurate, avoids incorrect claims, and stays true to the source material.
- **Language** – Choose the section that is clearer, more grammatically correct, and easier to read.
- **Overall Ranking** – Choose **the related work section you prefer** overall out of the two given.

G.4 Additional Notes

- Every **three consecutive tasks** share the same target paper; only the **two competing related works** change.

H Annotation Interface

The figure 7 shows the top portion of the annotation interface, specifically focusing on the statement annotation task. The layout presents the target paper on the left alongside two competing related work sections. Below these texts, the interface provides a dedicated area for the comparative evaluation aspects, followed by the full content of all cited papers for easy reference.

H.1 Evaluator Models

To ensure a comprehensive and robust evaluation of automated fact-checking capabilities, we employed a diverse suite of Large Language Models (LLMs) to act as judges. Our selection spans state-of-the-art proprietary models accessed via official APIs and high-performing open-weight models deployed locally. For the proprietary suite, we evaluated the latest iterations of the GPT family (GPT-5.4 and GPT-5.4-mini) and the Gemini family (`gemini-3-1-pro-preview` and `gemini-3-1-flash-lite-preview`).

For the open-weight suite, we utilized the `vLLM` framework to optimize local inference throughput. We evaluated the 70-billion parameter Llama 3.3 (`meta-llama/Llama-3.3-70B-Instruct`) and the 31-billion parameter Gemma 4 (`google/gemma-4-31B-it`). Furthermore, to investigate the computational trade-offs of quantization and chain-of-thought generation, we included an AWQ-quantized variant of Gemma 4 (`QuantTrio/gemma-4-31B-it-AWQ`). Notably, because the Gemma 4 architecture natively generates explicit reasoning traces, we evaluated this quantized checkpoint with its reasoning generation explicitly disabled. This allowed us to assess whether removing the computational overhead of chain-of-thought reasoning, combined with quantization, degraded the model's strict fact-checking performance. The complete list of evaluator models and their corresponding endpoints or repositories is detailed in Table 4.

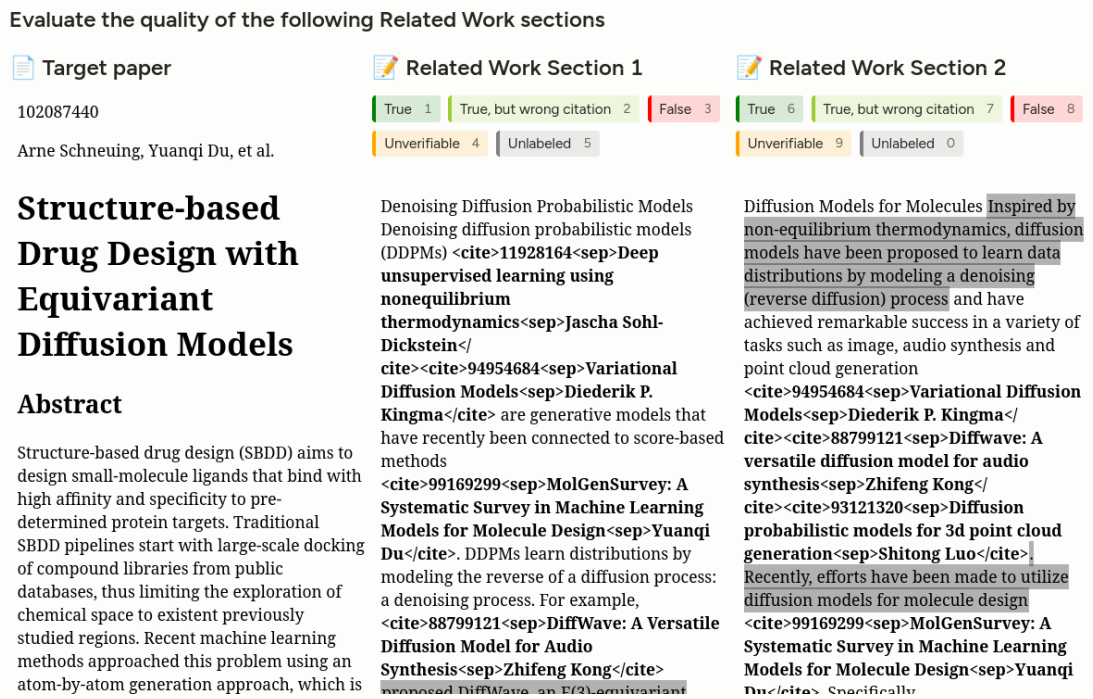


Figure 7: Annotation interface

Family	Specific Checkpoint / Version	Access Type
GPT	GPT-5.4	Prop. API
	GPT-5.4-mini	Prop. API
Gemini	`gemini-3-1-pro-preview`	Prop. API
	`gemini-3-1-flash-lite-preview`	Prop. API
Llama	`meta-llama/Llama-3.3-70B-Instruct`	Open (vLLM)
Gemma	`google/gemma-4-31B-it`	Open (vLLM)
	`QuantTrio/gemma-4-31B-it-AWQ`	Open (vLLM)

Table 4: Details of the LLM evaluators used in the statement-level factuality assessment. Open-weight models were executed locally using the vLLM engine, while proprietary models were accessed via their respective official APIs.

Aspect	κ
Preference	0.5099
Relevance	0.5452
Language	0.1707
Faithfulness	0.4301
Statement (all)	0.6169
Statements (randomly selected)	0.6556
Statements (randomly selected adversarial)	0.4368

Table 5: Inter-annotator agreement. We use quadratic weighted Cohen's κ for preference-based judgments (Preference/Relevance/Faithfulness/Language) because annotators could select *no preference*. We use standard Cohen's κ for statement labels (Statements).

I Inter-Annotator Agreement Details

We provide inter-annotator agreements in Table 5.

J Statement Level Evaluation

We focus on fine-grained statement-level faithfulness, as it yields higher inter-annotator agreement (Table 5) and serves as the foundation for our LLM judge meta-evaluation.

Initial fine-grained annotations yielded high rates of *Unverifiable* statements (41.3% for humans, 26.5% for GPT-4o-mini, 22.9% for PRIMERA). To address this, we conducted a meta-annotation round for samples lacking consensus between human annotators and the original statement extraction model. A single annotator, different from the previous two, reviewed these cases with the original annotations and the model-provided evidence, significantly decreasing the *Unverifiable* rate (Table 6). This suggests the initial round missed evidence due to the high cognitive load of scanning large text volumes. We provide detailed analyzes of unverifiable statements for human related works in Appendix K. Consequently, the initial high proportion of unverifiable claims likely influenced the coarse-level document ratings. The adversarial subset also consistently exhibited higher *False* and *Unverifiable* rates than the random subset, highlighting the difficulty of verifying these edge cases.

Overall, GPT-4o-mini beats human-written related work in factuality, aligning with Pu et al. (2023), who similarly observed high human hallu-

Model	N	True	W. Cite	False	Unver.
All Statements					
PRIMERA	140	58.6	5.7	30.7	**5.0**
Human	121	73.6	**0.8**	6.6	19.0
GPT-4o-mini	147	**78.2**	2.7	**5.4**	13.6
Randomly Selected					
PRIMERA	115	61.7	5.2	28.7	**4.4**
Human	110	76.4	**0.9**	3.6	19.1
GPT-4o-mini	116	**82.8**	2.6	**2.6**	12.1
Adversarial Subset					
PRIMERA	25	44.0	8.0	40.0	**8.0**
Human	11	45.5	**0.0**	36.4	18.2
GPT-4o-mini	31	**61.3**	3.2	**16.1**	19.4

Table 6: Fine-grained factuality evaluation. Metrics (aside from the total statement count, N) are reported as percentages (%). "W. Cite" denotes statements that are true but have the wrong citation, and "Unver." denotes unverifiable statements. The best performing model (highest True and lowest False) is in bold.

cination rates.

K Analysis of Unverifiable Statements in Human-Written Summaries

Because the proportion of unverifiable statements in human-authored summaries remained the highest even after the meta-annotation round, we conducted a deeper manual analysis of these statements. For this analysis, we reused the same subset of statements evaluated during the meta-annotation phase.

We found that 23 out of 121 statements (19.0%) in the human-written summaries were classified as unverifiable. Upon closer inspection of these 23 statements, we identified three primary causes for unverifiability:

First, 7 statements (30.4%) were unverifiable due to missing citations [17]. Second, 6 statements (26.1%) were flagged as unverifiable strictly due to automated parsing errors during extraction. Finally, 10 statements (43.5%) contained information or claims that were fundamentally ungrounded in the source material.

This also reflects the quality of the processing pipeline: only 6 out of 121 statements were identified as problematic due to the pipeline itself, corresponding to a success rate of approximately 95%.

[17]This includes instances where the relevant source was omitted entirely, or cited elsewhere in the target paper outside of the related work section.

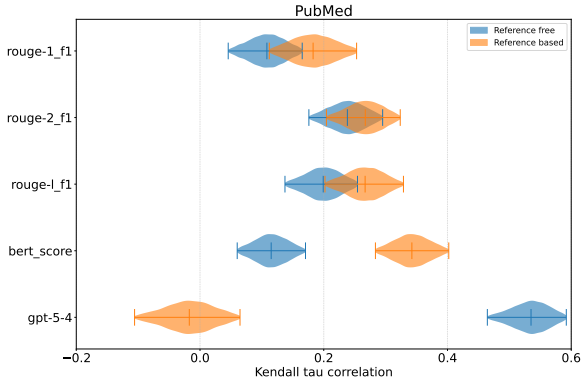


Figure 8: Violin plot showing 95% confidence interval of Kendall's τ correlations of automatic metrics with human judgment on LongEval's PubMed dataset.

L Comparison to LongEval

Here we provide Figure 8 showing LongEval results.

M Meta Evaluation - Detailed results

Table 7 provides statement-level meta-evaluation F1 scores for judges on the pre-extracted statements (Prompt 2, Appendix P). Gemini 3.1 Pro dominates, closely followed by Gemma 4 (31B). These LLMs outscore human annotators overall, highlighting the significant cognitive load of manual verification. We initially hypothesized that the poor performance of GPT models was due to selection bias, as the adversarial statements were mined using GPT-family models. However, detailed analysis reveals that GPT models also underperform on the non-adversarial subset, largely because they frequently misclassify *True, but wrong citation* and *Unverifiable* labels.

We also analyzed output formatting errors and found that 2.7% of all statements, created during evaluation of our experiments in Table 2, are missing the classification field. Nearly all of this (81% of affected statements) comes from just 26 out of 56 340 generations (0.05%) in which the judge model entered a degenerate repetition loop. The remaining, much more common failure affects 2 755 generations (4.9%) where exactly one statement was truncated right before its classification field, without any looping.

M.1 Note on Coarse Transformation

When aggregating statement-level classifications into coarse-level faithfulness scores, we observed that a small fraction of samples were associated

Evaluator Model	Subset	Class-Specific F1 Scores				Macro F1
		True	W. Cite	Unverifiable	False	
Human 82	All ($N = 408$)	0.74	0.72	0.46	0.57	0.62
	Random ($N = 341$)	0.76	0.74	0.45	0.49	0.61
	Adversarial ($N = 67$)	0.59	0.67	0.49	0.73	0.62
Human 83	All	0.76	0.42	0.43	0.46	0.52
	Random	0.76	0.40	0.41	0.47	0.51
	Adversarial	0.75	0.50	0.50	0.44	0.55
GPT-5.4	All	0.69	0.16	0.45	0.54	0.46
	Random	0.72	0.16	0.51	0.53	0.48
	Adversarial	0.45	0.19	0.19	0.56	0.35
GPT-5.4-mini	All	0.69	0.09	0.29	0.40	0.37
	Random	0.71	0.09	0.35	0.38	0.38
	Adversarial	0.57	0.10	0.00	0.45	0.28
Gemini 3.1 Pro	All	0.89	0.35	0.65	0.74	**0.66**
	Random	0.90	0.30	0.73	0.76	**0.67**
	Adversarial	0.79	0.55	0.33	0.70	**0.59**
Gemini 3.1 Flash-Lite	All	0.89	0.28	0.32	0.60	0.52
	Random	0.91	0.29	0.36	0.61	0.54
	Adversarial	0.79	0.27	0.17	0.58	0.45
Llama 3.3	All	0.82	0.12	0.17	0.36	0.37
	Random	0.85	0.15	0.13	0.33	0.36
	Adversarial	0.65	0.00	0.27	0.43	0.34
Gemma 4 (31B) Reasoning	All	0.88	0.32	0.65	0.70	0.64
	Random	0.89	0.31	0.72	0.72	0.66
	Adversarial	0.76	0.38	0.38	0.65	0.54
Gemma 4 (31B) Quant. (No Reas.)	All	0.88	0.23	0.58	0.64	0.58
	Random	0.89	0.22	0.62	0.66	0.60
	Adversarial	0.76	0.25	0.38	0.59	0.49

Table 7: Detailed performance breakdown of evaluator models across the All, Random, and Adversarial meta-annotated statement subsets. Class-specific F1 scores highlight model calibration on minority classes such as "True, but wrong citation" (W. Cite) and "Unverifiable".

with no statements. Because this proportion is negligible (e.g., under 0.1% for the Gemma model), we decided to treat these specific samples as fully faithful.

N Annotator Recruitment and Compensation

Our human evaluation involved three annotators, all recruited directly from the authors' institutional network (no crowdsourcing platforms were used). The two primary annotators were students in Computer Science, both with completed Bachelor's degrees in Electrical Engineering. The third annotator, responsible for the meta-annotation round was a PhD candidate.

The primary annotators were paid approximately 185 CZK/hour (net wage), roughly 1.6× the net minimum hourly wage (~113 CZK) close to the national net hourly median (~217 CZK). The PhD candidate participated on a voluntary basis as part of ongoing collaboration on the project. All annotators were informed in advance about the nature of the task.

O Licensing and Intended Use

The dataset contains open-access papers obtained from CORE and SemanticScholar corpora. These corpora contain third-party content and materials, such as open-access works from publicly available sources. In addition to the licenses of those organizations (ODC-By, CC BY-NC), any underlying Third Party Content may be subject to separate license terms by the respective third party owner. We made the best effort to provide identifiers (title, authors, year, DOI, or SemanticScholar ID) of collected papers to allow the user of this dataset to check the license.

The OARelatedWork dataset and the OAPapers corpus are provided solely for research purposes, including but not limited to experimentation with

machine learning models and data analysis.

P Prompts

This section provides prompts for used LLMs.

Prompt 1: Statement Extraction Prompt for Human Evaluation

Role: system

```
You are a scientific research assistant. Your role is to review factual statements within the related work sections of
↪  scientific papers and verify their accuracy using only information from the supplied cited papers and the target paper
↪  for which the related work was written.
Definition: A factual statement is objective, usually no longer than one sentence, and can be verified as true or false. Do
↪  not include subjective opinions or personal experiences.

Format of texts:
- Paragraphs are in <p>...</p> tags.
- Headings are in <h1>...</h1>, <h2>...</h2>, etc. tags.
- Citations are in <cite>ID<sep>paper title<sep>first author</cite> tags. For citations to unknown papers the cite tag
↪  contains only <cite>UNK</cite>.
- References to figures and tables are in <ref>figure|table|UNK/ref> tags.
When extracting text leave the tags as is.

Begin with a concise checklist (3-7 bullets) of your approach; keep items conceptual, not implementation-level.
Instructions:
- For each provided statement, determine if it can be verified exclusively using the provided cited papers and the target
↪  paper.
- If a statement cannot be verified due to lack of supporting evidence from these sources, mark it as not factual and set
↪  the evidence array to empty.
- If there are multiple statements, process each statement in order and return a separate JSON object for each.
- For statements that are malformed or lack citations, set 'is_true' to false and provide an empty 'evidence' array.
Evidence:
- Each item in the 'evidence' array must be an object containing a 'source' field (the ID of the cited paper or
↪  'target_paper') and an 'excerpt' field (a supporting passage from the source).
- The 'evidence' array can have zero or more entries; include all relevant excerpts if multiple sources support your
↪  decision.
After processing each statement, briefly validate that your factual assessment and evidence selection are appropriately
↪  grounded in the provided sources. If validation fails or is ambiguous, self-correct before finalizing your output.
Output Format:
Return a JSON array. Each element corresponds to a processed input statement and the output follows this schema:
[
{
"statement": "repeat the original statement here, verbatim",
"evidence": [
{
"source": "ID of cited paper or 'target_paper'",
"excerpt": "text excerpt from the source supporting your decision"
}
// ...add more evidence objects as needed
],
"is_true": true | false,
"confidence": confidence score from 0 (completely unsure) to 1 (certain)
}
// ...more elements as needed
]
If no evidence is found or the statement is malformed, set the 'evidence' array empty.
```

Role: user

```
Cited papers:
{% for cp in cited_papers %}
{{cp}}

{% endfor %}

Target paper:
{{target_paper}}

Related work section:
{{rw}}

Please identify and verify factual statements in the related work section based on the cited papers and target paper.
```

Prompt 2: Statement Classification Prompt for Meta-Evaluation

Role: system

```
Role: You are an expert scientific reviewer fact-checking the "Related Work" section of academic papers.

Task: You are given a factual statement from the provided "Related Work" section; verify it against the provided reference material (the rest of the paper and the cited papers). Do not use the Related Work section itself as reference material for verification.

Constraints:
- KNOWLEDGE BOUNDARY: Base your decisions strictly and solely on the provided input text. Do not use external knowledge or training data.

Verification Categories (For your internal processing):
1. "True": The statement is fully supported by the reference material AND the specific citation provided is correct.
2. "True, but wrong citation": The statement is factually supported by the reference material, but the specific citation attached to it does not support it.
3. "Unverifiable": The provided reference material does not contain enough information to prove the statement true or false.
4. "False": The statement is directly contradicted by the reference material.

Instructions:
1. Read the context of the statement in the "Related Work" section to understand its meaning and the specific citation attached to it. The statement is highlighted in the provided "Related Work" section with <statement> tags.
2. Search for evidence in the reference material that can be used to classify the statement into one of the four Verification Categories. The evidence must be directly quoted text from the reference material that supports or contradicts the statement.
3. Write a rationale (maximum two sentences) explaining your final decision.
4. Make a final classification into one of the four Verification Categories.

Output Format:
Output the following information in a structured format (e.g., JSON):
{
  "evidence": [
    "direct quote from the reference material that supports or contradicts the statement"
  ],
  "rationale": "a brief explanation of the reasoning behind the classification",
  "classification": "one of the four Verification Categories (True, True but wrong citation, Unverifiable, False)"
}
```

Role: user

```
Please review the following factual statement from the "Related Work" section and evaluate it against the provided reference materials.

<reference_material_rest_of_paper>
{{ txt_target_paper | truncate(tokenizer_name="casperhansen/llama-3.3-70b-instruct-awq", number_of_tokens=30000, direction="right", backend="transformers") }}
</reference_material_rest_of_paper>

<reference_material_citations>
{{ txt_cited_papers | truncate(tokenizer_name="casperhansen/llama-3.3-70b-instruct-awq", number_of_tokens=90000, direction="right", backend="transformers") }}
</reference_material_citations>

<statement>
{{ text }}
</statement>

<related_work>
{{ txt_rw[:start] }}<statement>{{ txt_rw[start:end] }}</statement>{{ txt_rw[end:] }}
</related_work>
```

Prompt 3: The Judge Prompt

Role: system

```
Role: You are an expert scientific reviewer fact-checking the "Related Work" section of academic papers.

Task: Extract all factual statements from the provided "Related Work" section, verify them against the provided reference
↪  material (the rest of the paper and cited papers).

Constraints:
- KNOWLEDGE BOUNDARY: Base your decisions strictly and solely on the provided input text. Do not use external knowledge or
↪  training data.

Definitions:
- Factual Statement: An objective claim (typically one sentence or less) devoid of subjective opinion, capable of being
↪  verified against the reference material.

Verification Categories (For your internal processing):
1. "True": The statement is fully supported by the reference material AND the specific citation provided is correct.
2. "True, but wrong citation": The statement is factually supported by the reference material, but the specific citation
↪  attached to it does not support it.
3. "Unverifiable": The provided reference material does not contain enough information to prove the statement true or false.
4. "False": The statement is directly contradicted by the reference material.

Instructions:
1. Systematically identify every factual statement in the "Related Work" section.
2. Search for evidence in the reference material that can be used to classify each statement into one of the four
↪  Verification Categories. The evidence can only be directly quoted text from the reference material that supports or
↪  contradicts the statement.
3. Write a rationale (maximum two sentences) explaining your final decision.
4. Make a final classification for each statement into one of the four Verification Categories.

Output Format:
For each factual statement, output the following information in a structured format (e.g., JSON):
{
  "statements": [
    {
      "text": "exact text of the factual statement extracted from the Related Work section",
      "evidence": [
        "direct quote from the reference material that supports or contradicts the statement"
      ],
      "rationale": "a brief explanation of the reasoning behind the classification",
      "classification": "one of the four Verification Categories (True, True but wrong citation, Unverifiable, False)"
    },
    ...
  ]
}
```

Role: user

```
Please review the following "Related Work" section and evaluate its factual statements against the provided reference
↪  materials.

<reference_material_rest_of_paper>
{{ target_paper }}
</reference_material_rest_of_paper>

<reference_material_citations>
{{ cited_papers }}
</reference_material_citations>

<related_work>
{{ summary }}
</related_work>
```

Prompt 4: LLM baselines prompt when all content is used (PART 1)

Role: system

```
Write the related work section for the paper described bellow and given references.

Format instructions:
  Write paragraphs on separate lines.
  Use in-text citations for references using following format: <cite>ID<sep>TITLE<sep>FIRST_AUTHOR</cite>.
  {% if structured %}Use subsections for different topics. Every subsection has a title on a separate line with a # prefix.
  ↪  For the first subsection, use ##, for the second, ###, etc.{% else %}Do not use subsections.{% endif %}
  Write just the content of the related work section. This means that you are not supposed to write the title of the section
  ↪  or the references section.


An example of related work section:

{% if structured %}
## 2.1 Load balancing
Many research have been done to address the issue of load balancing in distributed systems. They are typically classified
↪  into static and dynamic ones <cite>30331579<sep>Comparison of dynamic and static load-balancing strategies in
↪  heterogeneous distributed systems<sep>Y Zhang</cite>. Static policies determine job assignments based on information
↪  about the average behavior of the system. They are simple, efficient, and easy to implement, but cannot adapt to the
↪  changing workload in the system. To respond to the fluctuation of workload over time, dynamic load balancing policies
↪  are proposed. Ref. <cite>18652554<sep>A dynamic load-balancing policy with a central job dispatcher (LBC)<sep>H C
↪  Lin</cite>, for example, gives a policy with a central job dispatcher, which periodically collects load information at
↪  system nodes and makes decisions for job transformation. They are more complex than static policies, but exhibit better
↪  performance. These load balancing policies view system nodes as a collection of common resources for arriving jobs. This
↪  makes them inapplicable to the scenario we are concerned about, in which a query should be processed by a particular
↪  server where the queried data are located. Refs. <cite>10822818<sep>Data distribution algorithms for load balanced
↪  fault-tolerant web access<sep>B Narendran</cite><cite>5163968<sep>Approximation algorithms for data distribution with
↪  load balancing of web servers<sep>L C Chen</cite><cite>5700028<sep>Approximate algorithms for document placement in
↪  distributed web servers<sep>S Tse</cite> address the issue of data allocation; however they are typically designed to
↪  allocate web documents among a cluster of web servers, where the document size and HTTP connection numbers are main
↪  concerns.
A more relative work is data distribution in parallel storage systems. In these systems, horizontal data partitioning has
↪  been commonly used to distribute data among system nodes. There are typically three strategies for horizontal
↪  partitioning: round-robin, hash, and value-range partitioning. These partitioning strategies each has advantages and
↪  disadvantages, but they do not take into consideration the access pattern on data, which would be a problem when the
↪  pattern is not a uniform distribution. Dynamic data reallocation needs to effectively adapt to changes of access
↪  patterns. When access frequencies of data items lead to unbalanced workload on system nodes, data migration should be
↪  performed to balance workload. The simple data skew handling method in <cite>18421049<sep>Fat-Btree: An
↪  update-conscious parallel directory structure<sep>H Yokota</cite> balances the storage space for data, but does not
↪  guarantee balanced data access load across system nodes. Similarly most access load skew handling methods do not
↪  consider data balancing <cite>4657232<sep>Towards self-tuning data placement in parallel database systems<sep>M L
↪  Lee</cite><cite>111471344<sep>A fast convergence technique for online heat-balancing of Btree indexed database over
↪  shared-nothing parallel systems<sep>H Feelifl</cite>. In <cite>12316292<sep>Adaptive lapped declustering: A highly
↪  available data-placement method balancing access load and space utilization<sep>A Watanabe</cite>, a data-placement
↪  method was proposed to balance both access load and storage space of data, but it is designed for parallel storage
↪  system, and focuses on achieving availability and scalability for parallel storage configuration.
## 2.2 Market mechanism
During the past few years, market mechanism has gained much interest in both the economics and computer science literature.
↪  Market-based techniques <cite>3303225<sep>Characterizing effective auction mechanisms: insights from the 2007 tac
↪  market design competition<sep>J Niu</cite> are applying themselves to resource allocation, often through the creation
↪  of artificial market. This is due to the fact that market mechanisms, when well designed, can achieve desired economic
↪  outcomes like high allocated efficiency. Auctions are a subclass of market mechanisms that have received particular
↪  attention. Double auction <cite>117078532<sep>The double auction institution: A survey<sep>D Friedman</cite>, on which
↪  our method is based, is one type of auction where both buyers and sellers can make offers. There is a wide range of
↪  literature published on double auctions (e.g., <cite>1372607<sep>Market power and efficiency in a computational
↪  electricity market with discriminatory double-auction pricing<sep>J Nicolaisen</cite><cite>124078139<sep>Pricing in
↪  agent economies using multi-agent Q-learning<sep>G Tesauro</cite>).
Connecting load balancing to the market mechanism in the distributed environment is a new direction of research. In our
↪  approach, we will show how to integrate the ideas of the artificial market into solving load balancing challenges in the
↪  cloud-based environment.
{% else %}
```

Prompt 5: LLM baselines prompt when all content is used (PART 2)

```
Recent years have witnessed the huge growth of the Internet, and the emergence of new applications that require guaranteed
↪  service levels. Real-time consumer applications, such as streaming live video and online gaming have introduced new
↪  traffic characteristics and imposed new requirements on network performance and resource availability. So far, the
↪  Internet has offered only best-effort services. All traffic is delivered in the best possible way with no guarantees
↪  given to any type of traffic. However, today there are more applications that demand service guarantees in order to
↪  function properly. Supporting these new types of applications requires more sophisticated routing mechanisms for path
↪  selection and for better network resource utilization. The concept of Quality of Service (QoS) has been recently
↪  addressed extensively in the literature as a means of providing the required service level for demanding applications
↪  <cite>13100834<sep>Survey of qos routing<sep>P Paul</cite>.
QoS routing is a major component of the QoS paradigm; it consists of a set of routing related mechanisms that guarantee to
↪  provide the service level required by traffic flows. In QoS routing, paths for flows are selected based upon knowledge
↪  of the resource availability, referred to as either link-state information or network-state information, held by the
↪  network nodes and the QoS requirements of the flows.
QoS routing selects a path that has sufficient resources to accommodate the QoS requirement of a given flow
↪  <cite>5747745<sep>A Framework for QoS-based Routing in the Internet<sep>R N E Crawley</cite>. In QoS routing, some
↪  knowledge regarding the global link state is required by each network node in order to take routing decisions.
This knowledge can be obtained through a periodic exchange of the network state among the network nodes. Using link-state
↪  updates, each node constructs a global view of the network state and selects the best path for a flow based on this
↪  global view of the network. Global QoS routing performs well when each source node has a reasonably accurate view of the
↪  entire network state. However, the network state changes with each flow arrival or departure and maintaining an accurate
↪  network state becomes impractical, due to the rapid flow dynamics. Moreover, the exchange of link states among nodes
↪  introduces prohibitive communication and processing overheads. As a result of inaccurate state information, routing may
↪  suffer degraded performance and instability <cite>UNK<sep>Quality of Service in IP networks<sep>G
↪  Armitge</cite><cite>9587075<sep>An Overview of Quality-of-Service Routing for the Next Generation HighSpeed Networks:
↪  Problems and Solutions<sep>S Chen</cite>. Consequently, an alternative routing approach that can circumvent the
↪  problems associated with global state information is sought.
{% endif %}
```

Role: user

```
Write the related work section for the paper described bellow and given references.

The related work section should have around {{summary_length_words}} words.

Description of the paper:{% set truncated_queries = queries | truncate_group(max_length=22000) %}
Title:
{{truncated_queries[0]}}

Abstract:
{{truncated_queries[1]}}

Rest of the paper:
{{truncated_queries[2]}}
```

Role: assistant

```
Please provide references that should be used in the related work section.
```

Role: user

```
Sure, here are the references:

{% for paper in inputs | truncate_group(max_length=80000) %}
{{paper}}

{% endfor %}

Use them to write the related work section.
```