ACL-Verbatim: hallucination-free question answering for research

Gábor Recski[1,2], Szilveszter Tóth[2], Nadia Verdha[1], István Boros[2], Ádám Kovács[2]
[1]TU Wien, [2]KR Labs
Correspondence: recski@krlabs.eu

Abstract

Researchers need efficient and reliable methods for collecting high-quality information from trusted sources, but modern tools for AI-assisted research still suffer from the tendency of Large Language Models (LLMs) to produce factually inaccurate or nonsensical output, commonly referred to as hallucinations. We apply the extractive question answering system VerbatimRAG (Kovacs et al., 2025) to research papers in the ACL Anthology[1], directly mapping user queries to verbatim text spans in retrieved documents. We contribute a novel ground truth dataset for the task of mapping user queries to relevant text spans in research papers, and use it to train and evaluate a variety of extractive models. Human annotation is performed by NLP researchers and is based on synthetic user queries generated using a custom pipeline based on the ScIRGen methodology (Lin et al., 2025), paired with chunks of research papers retrieved by VerbatimRAG. On this benchmark, a 150M-parameter ModernBERT token classifier trained on silver supervision from our pipeline achieves the best word-level F1 (53.6), ahead of the strongest evaluated LLM extractor (48.7).

1 Introduction

Researchers rely on scientific literature as a trusted source of information, but finding the relevant evidence in large paper collections remains difficult. Modern AI tools, especially those based on large language models (LLMs), offer a substantial increase in the efficiency of information search but introduce major risks to both individual users and organizations. Question Answering using LLMs lacks transparency and reliability. Even retrieval-augmented generation (RAG) systems (Gupta et al., 2024), which use a combination of document retrieval and generative AI, are prone to most major issues of LLMs, including a tendency to produce factually inaccurate, irrelevant, or nonsensical output, commonly referred to as hallucinations (Huang et al., 2025). Answers provided by LLMs cannot be trusted unless they are independently fact-checked, yet such verification remains a tedious process and is often omitted due to "algorithm appreciation, where people tend to prefer algorithmic judgment over human judgment, even when the algorithm's processes are not fully transparent" (Logg et al., 2019)

We present *ACL-Verbatim*, an application of the VerbatimRAG framework for extractive question answering (Kovacs et al., 2025) to the task of question answering from research papers in the ACL Anthology. While we contribute an end-to-end RAG system, complete with document preprocessing, indexing, and retrieval, our main focus is *extraction*, the task of identifying spans in retrieved text chunks that are most useful for satisfying the information need conveyed by the user query. It is this step that differentiates VerbatimRAG from other RAG frameworks and which enables question answering without hallucinations. In order to train and evaluate models on this task, we also create a pipeline for the automatic generation of search queries and perform manual annotation of text chunks to create a small ground truth dataset of 100 query-chunk pairs. Finally, we show that fine-tuning a compact extraction model on silver data generated by a strong LLM yields the best word-level F1 on our benchmark while using far fewer parameters than the evaluated LLM extractors.

2 Related work

All question answering systems that allow an LLM to generate the final answer are prone to producing output that is factually incorrect, inconsistent with the provided evidence, or nonsensical, a phenomenon that is commonly referred to as hallucination (Kaddour et al., 2023; Huang et al., 2025;

[1] https://aclanthology.org/

Proceedings of the 4th Workshop on Artificial Intelligence for Scientific Publications, pages 23–33
November 10, 2026 ©2026 Association for Computational Linguistics

Ji et al., 2023). Retrieval Augmented Generation (RAG) systems reduce intrinsic hallucinations by grounding the model in external sources, but extrinsic hallucinations can still occur due to LLMs' tendency to override retrieved information with their own prior "knowledge". RAG models continue to hallucinate (Niu et al., 2024), limiting their use in complex and high-risk domains such as medical, legal, or financial question answering (Lozano et al., 2023; Magesh et al., 2024). A range of methods have recently been proposed for hallucination detection. Frameworks such as RAGAS (Es et al., 2024) and ARES (Saad-Falcon et al., 2024) rely on specialized LLMs for large-scale hallucination detection but are not suitable for real-time prediction. Other LLM-based methods include approaches that use stochastic sampling (Manakul et al., 2023) or multi-step verification (Friel and Sanyal, 2023). Classifier models trained on hallucination datasets such as RAGTruth (Niu et al., 2024) include RAG-HAT (Song et al., 2024), RAGHalu (Zimmerman et al., 2024), and LettuceDetect (Ádám Kovács and Recski, 2025). Approaches to hallucination detection that investigate connections between responses and source documents include Luna (Belyi et al., 2025) and FACTOID (Rawte et al., 2024). Despite these recent efforts, hallucinations in RAG systems continue to limit the applicability of LLMs in real-world question answering tasks.

LLMs also suffer from lack of explainability, reducing both accountability and user trust. Mechanisms for generating post-hoc explanations of neural networks' predictions are notoriously unreliable, and intuitive but wrong self-explanations offered by LLMs create additional risk by inflating users' perception of their trustworthiness (Madsen et al., 2024; Chen et al., 2024b). These risks are inherent to systems that allow neural language models to generate the final output presented to users, even when these models have been specialized for the domain of academic research (Beltagy et al., 2019; Taylor et al., 2022; Viswanathan et al., 2023).

VerbatimRAG (Kovacs et al., 2025) is an open-source RAG framework that tackles the issue of hallucinations by taking an extractive approach to retrieval-augmented question answering that only returns text spans that are taken verbatim from source documents. VerbatimRAG combines standard retrieval with extraction, the task of highlighting the parts of some input text that are relevant for answering some user query, for which the framework offers multiple approaches, including LLMs prompted for the extraction tasks as well as smaller models fine-tuned for the extraction/highlighting task, such as Provence (Chirkova et al., 2025) or the Zilliz Semantic Highlighter (Zhang and Chen, 2026). Our experiments as well as our newly contributed dataset rely on the ACL Anthology[2], a public resource that has served as the basis of dozens of research datasets over the past decades (Bollmann et al., 2023), including large-scale corpora such as NLP Scholar (Mohammad, 2020), NLP-Explorer(Parmar et al., 2020), and the most recent ACL-OCL corpus (Rohatgi et al., 2023), each of which provides valuable additional metadata for publications and enables advanced analysis of NLP research.

3 Corpus creation

In this section we describe the corpus creation process, including data collection, preprocessing, segmentation, the generation of synthetic queries, as well as the human annotation process. Our pipeline is designed to allow for incremental updates of our dataset based on updates of the ACL anthology, detailed instructions are provided in the `acl-verbatim` repository[3]. The version of the dataset that served as the basis for the annotation and evaluation described in this paper is based on the state of the ACL Anthology in February 2026.

3.1 Data collection and preprocessing

The ACL Anthology (Gildea et al., 2018; Bollmann et al., 2023) is a public library of over 120,000 research papers from the domains of computational linguistics and natural language processing. Metadata as well as full-text PDFs of papers are distributed under permissive licenses (CC BY 4.0 as of 2016) and programmatic access is provided via a GitHub repository[4] and a Python library. We use these utilities to process all PDF files and to extract all paper metadata. Downloading, filtering, and preprocessing of papers was based on metadata extracted from the ACL Anthology on February 26, 2026. Entries for 120 034 papers were processed, out of which 114 567 were mapped to PDFs for further processing. The remaining 5k papers are hosted by third-party publishers and not covered by the permissive license of the ACL Anthology, these entries were discarded to maintain the flexible terms of the final dataset.

[2] https://aclanthology.org/
[3] anonymized
[4] https://github.com/acl-org/acl-anthology/

PDFs are converted to markdown format using the open-source `docling` library[5], resulting in 114 475 markdown files (a total of fewer than 100 papers were skipped due to a variety of docling errors that were not further investigated). Docling's DocumentConverter is invoked using default settings. All text-based content including headers, lists, tables, figure captions etc. are rendered in markdown, while other figures and some formulas are replaced by placeholder text indicating that some content has been discarded. We release this dataset of markdown files under a CC-BY 4.0 license on Huggingface[6].

Markdown documents are indexed using the VerbatimRAG library described in Section 2. For segmenting papers we implement a custom chunking strategy developed specifically for markdown-formatted research papers. This involves parsing section structure, segmenting papers along section boundaries, and prefixing each text chunk section and subsection titles to improve retrieval performance. The markdown chunker also prevents tables and code blocks from being split, and controls the minimum and maximum size of chunks, which we set to 500 and 5000 characters, respectively. Chunks are then indexed both for full text search and for dense vector search using the `granite-embedding-english-r2` embedding from IBM[7] (Awasthy et al., 2025).

3.2 Query generation and human annotation

This section describes the steps of creating a ground truth dataset mapping user queries to text spans in retrieved papers relevant for answering these queries. As a first step we create a sample of 333 papers in the ACL Anthology, randomly choosing from all English-language papers. Then we retrieve indexed chunks for these papers from the ACL-Verbatim index and randomly choose a single chunk for each paper. We then generate 3 synthetic queries for each chunk, following the ScIRGen methodology (Lin et al., 2025). This two-step process involves prompting an LLM to generate a list of question types that could be answered by a given paragraph, then using in-context learning for each question type to generate questions. We extend this pipeline by a third step that converts long and linguistically sophisticated questions to shorter and more fragmented queries that are more characteristic of real-world user queries. An end-to-end example is presented in Appendix A, full prompts are reproduced in Appendix D.

Generated queries are used to retrieve chunks from the VerbatimRAG index (see Section 3.1), and the top 5 chunks per query are used as input for the annotation task. Annotators then perform tasks for pairs of query and chunk. First, each chunk is annotated for relevance using a binary label. Chunks must be marked `relevant` if and only if the annotator considers the chunk to be relevant for satisfying the information need conveyed by the search query. If a chunk is labeled `irrelevant`, no further annotation takes place[8]. For chunks considered relevant, annotators must also indicate (highlight) the span or spans of text within the chunk that are most relevant for answering the query. If a table or figure is considered relevant, annotators are instructed to highlight its caption. This way a sequential labeling task is defined, mapping each relevant text chunk to one or more continuous sequences of its tokens. Annotation is performed via an excel sheet, created programmatically from the output of the ACL-Verbatim system and postprocessed to create the JSON-formatted gold dataset. All components of the pipeline for query generation and annotation are published as open-source software on GitHub[9].

3.3 Annotation challenges

The core annotation task of determining the text fragments within a section of a research paper that are best suited to fulfill the information need conveyed by a search query is quite complex and raises several methodological issues. While the sampling process described above yielded a total of 906 queries, and retrieving the top 5 chunks for each produced 4530 query-chunk pairs, the human-annotated portion of our dataset contains only the first 20 queries and a total of 100 chunks. Annotation as well as the adjudication of differences among annotators was performed by authors of this paper, NLP researchers with some variety in their fields and level of experience. In this section we describe some of the key challenges we have encountered during the manual annotation process and discuss some implications for the extraction

[5] https://docling-project.github.io/docling/

[6] anonymized

[7] https://huggingface.co/ibm-granite/granite-embedding-english-r2

[8] Additionally, a small number of chunks were marked with a question mark to indicate that the relevance judgement doesn't make sense, this was the case in particular for chunks containing bibliography sections of papers. Such chunks were also not annotated further.

[9] anonymized

task underlying the `verbatim` approach to question answering. We illustrate issues with examples cherry-picked from the 20 queries in the manually annotated dataset.

A key challenge that is specific to the domain of academic research is that for many queries the annotation of the most relevant spans within a section of a paper requires considerable domain-specific expertise and careful consideration. For example, the (synthetic) query *parsing merge predicate sequence equivalence conditions*, which is the simplified version of the more elaborate (synthetic) question *What are the three conditions under which two instantiated sequences are considered equivalent by the parsing merge predicate?*, which in turn was generated based on a short section describing an algorithm in the paper *LR Recursive Transition Networks for Earley and Tomita Parsing* (Perlin, 1991). Retrieving the top 5 chunks from our index yields sections from 4 different papers. Perlin 1991 is not among them, but all of them are concerned with algorithms for syntactic parsing and all of them would appear to be potentially relevant based on their vocabulary. It is only by reading through and developing a basic understanding for each section that the annotators could make the first of two judgements, the binary decisions on whether these chunks are relevant at all and whether one should proceed with the extraction task to identify relevant spans. The final dataset categorizes two of the chunks as relevant, the introductory section of the 1989 EACL paper on *Parsing and Derivational Equivalence* (Hepple and Morrill, 1989) and a paragraph describing a core algorithm in the 2010 ACL paper *Dynamic Programming for Linear-Time Incremental Parsing* (Huang and Sagae, 2010). Further annotation mapped both of these chunks to spans that should be extracted (highlighted) in response to the search query, which in one case reduced the 4700 character algorithm description to a single sentence of 92 characters (1.96%) *"The key observation for dynamic programming is to merge 'equivalent states' in the same beam"*, while the introductory section of the other paper was mostly relevant and only a few sentences were omitted from the extraction, reducing 1902 characters to 1447 (76.08%). This example illustrates the diversity of extracted spans, the difficulty and subjectivity of both annotation steps, and the meticulous effort required to create even a small high-quality dataset.

4 Extraction experiments

We evaluate extractive models on the manually annotated benchmark introduced in the previous sections. The benchmark contains 20 synthetic queries paired with the top-5 retrieved chunks per query, yielding 100 query–chunk pairs in total. Of these, 47 chunks are annotated as relevant and contain 78 gold evidence spans, while the remaining 53 chunks are irrelevant and have no gold spans. We report extraction metrics on all 100 rows. Three families of extractive systems are compared. First, we evaluate LLM-based span extractors. These models receive a question together with the retrieved chunk and must return verbatim evidence spans. We evaluate Mistral Small 2603, Nemotron-120B-A12B, GLM-5, and Qwen 3.6 35B. For Mistral, Nemotron, and Qwen, we compare a default extraction prompt against a paragraph-oriented prompt designed to encourage broader evidence selection. Second, we evaluate extractive pruning and highlighting baselines. Zilliz Semantic Highlight (Zhang and Chen, 2026) selects relevant sentences or token spans from the chunk, while Provence (Chirkova et al., 2025) prunes irrelevant sentences from the context using a DeBERTa-v3 reranker-style architecture. Zilliz follows the same general token-scoring formulation as Provence, but is trained as a bilingual semantic-highlighting model on top of a BGE-M3 reranker backbone (Chen et al., 2024a). Provence operates with a native context budget of 512 tokens and internally splits longer chunks.

Finally, we also train a compact self-contained student model, for which we generate silver supervision from the ACL Anthology corpus using synthetic queries and retrieved chunks, following the steps described in Section 3.2. The current release is based on 2000 sampled papers, which yielded 5892 synthetic queries, 32480 raw silver query–chunk rows, and 23235 retained rows after filtering. The final split contains 20916 silver training rows and 2319 development rows in canonical form, and 20920 / 2319 tokenized training / development windows at the full 8192-token ModernBERT context. At this context length roughly every silver row fits in a single window; the total amount of silver supervision the student sees is approximately 10k positive rows with spans and 11k negative rows. Evidence density (span characters over chunk characters) is 11.7%, i.e. a roughly 1:8 token-level class imbalance.

Our student architecture is a query-conditioned token classifier over an 8192-token ModernBERT backbone, with binary token labels and sliding-window inference. The input is the concatenation of question and chunk, and the output is a binary evidence label per token, decoded into character spans. We compare two backbones: the vanilla `answerdotai/ModernBERT-base` MLM checkpoint and the `Alibaba-NLP/gte-reranker-modernbert-base` cross-encoder, which has been post-trained on query–passage relevance. The silver teacher is Qwen 3.6 35B with the paragraph-oriented extraction prompt. We train for 5 epochs with batch size 8 at learning rate 2×10^{-5}; the best checkpoint is selected by silver-dev token F1. Two post-processing steps are applied at inference: spans shorter than 10 characters are dropped, and neighboring spans separated by at most 20 characters are merged. These two rules remove token-level fragmentation in which the model emits a "shotgun" of short pseudo-spans around genuine evidence tokens. The student model based on the reranker backbone will be released as `acl-verbatim-modernbert`.

We propose several overlap-based metrics for comparing extracted text spans against the ground truth on all 100 query–chunk rows in the benchmark. Our primary method of evaluation and comparison is word-level precision and recall, which compares the sets of words covered by gold and extracted spans. We prefer this metric because it isolates a model's ability to highlight the right words in a piece of text and is not sensitive to whether span boundaries are correctly predicted and whether models (and annotators) prefer fewer and longer spans or many shorter ones. The best configurations for each model type were selected by comparing word-level F1 scores. Additionally, *Containment* measures whether a large enough part of predicted spans are contained by gold spans (e.g. *Containment @ 0.8* is the ratio of spans that are at least 80% covered by a gold span). Analogously, *Coverage @ 0.8* is the number of gold spans that are at least 80% covered by predicted spans, divided by the total number of gold spans. Overlaps between spans are measured at the character level.

The right choice of metrics for evaluating span extraction depends heavily on the nature of the extraction task. One of the most common span extraction tasks in natural language processing is Named Entity Recognition (NER), where system outputs are commonly evaluated using span-level precision and recall, which requires exact matches between span boundaries and will penalize even the smallest mismatch by considering it as both a false positive and a false negative. We argue for our choice of metrics with a simple example. Consider a sequence of 100 characters in a retrieved chunk of text that is annotated as containing two relevant spans of 45 characters each, with a 10 character long break between them. Then consider a system that predicts this entire span of 100 characters as relevant, i.e. its mistake is merging the 10 irrelevant characters with the 90 relevant ones. This prediction would achieve span-level precision and recall scores of zero, since it has not made any correct predictions. In contrast, the word-level precision and recall of the system are 0.9 and 1.0, respectively. Finally, the *Containment* @ t ratio is 1 for all $t < 0.9$ and 0 for $t \geq 0.9$, while *Coverage* @ t is 1 for any value of t, expressing alternative preferences in evaluating extraction.

In addition to the ACL-specialized model evaluated here, we release a multi-domain sibling model, `verbatim-rag-modern-bert-v2`[10], trained on `verbatim-spans`[11]. This dataset combines our ACL silver data with RAGBench (Friel et al., 2025), a large-scale benchmark of retrieval-augmented question answering examples across industry-oriented domains, and Squeez (Ádám Kovács, 2026), a task-conditioned tool-output pruning dataset built from coding-agent tool observations. We evaluate this generic model separately in its model card, including QASPER (Dasigi et al., 2021), a scientific QA benchmark over NLP papers, as an out-of-training-domain test set. On ACL gold the generic model reaches 46.3 word-level F1, compared to 30.1 for Zilliz Semantic Highlight and 34.4 for Provence, and it also outperforms both baselines on RAGBench, Squeez, and QASPER.

5 Results

Table 1 compares extractor models on the 100 rows of the manually annotated benchmark. The best Word-F1 is achieved by our reranker-initialized ModernBERT student (53.63), ahead of the strongest LLM extractors, GLM-5 (48.71), Mistral Small (46.94), and Qwen with the paragraph prompt (46.73), while using 3 to 4 orders of magnitude fewer parameters. The generic multi-domain

[10] anonymized
[11] anonymized

Model	*Word*			*Containment @ t*			*Coverage @ t*			Latency
	Prec.	Rec.	F1	1.0	0.8	0.5	1.0	0.8	0.5	
LLM extractors										
Mistral Small 2603[a]	40.41	55.99	46.94	43.86	46.24	49.73	17.95	39.74	57.69	0.78
Mistral Small 2603[b]	34.22	**73.03**	46.61	27.70	33.61	39.04	48.72	**65.38**	**75.64**	1.07
Nemotron-120B-A12B[a]	34.14	50.92	40.88	36.62	36.62	42.88	43.59	43.59	52.56	0.38
Nemotron-120B-A12B[b]	30.15	62.72	40.73	29.93	31.29	44.90	**58.97**	62.82	62.82	0.54
GLM-5[a]	44.50	53.80	48.71	**52.87**	**54.25**	**58.36**	20.51	34.62	53.85	1.04
Qwen 3.6 35B[a]	44.74	40.88	42.72	49.05	50.95	53.33	15.38	25.64	41.03	1.63
Qwen 3.6 35B[b]	39.43	57.35	46.73	34.78	36.96	47.83	44.87	51.28	61.54	1.20
Pruning / highlighting baselines										
Zilliz Semantic Highlight[c]	46.97	22.11	30.07	33.33	35.46	39.53	2.56	8.97	19.23	1.04
Provence[d]	27.58	45.70	34.40	21.56	22.84	29.23	16.67	21.79	41.03	2.40
Our students (150M params)										
acl-verbatim-modernbert [e]	**65.43**	45.43	**53.63**	27.27	31.47	48.22	35.90	37.18	42.31	0.47
verbatim-rag-modern-bert-v2 [f]	63.00	36.58	46.29	30.30	31.82	46.97	26.92	32.05	38.46	0.40

Table 1: Extractor results on the gold benchmark (100 query–chunk pairs: 47 relevant and 53 irrelevant). See Section 4 for metric definitions. Values are percentages, except latency, which is seconds per row. Latencies for Zilliz, Provence, and ModernBERT were measured on CPU.

[a] Default extraction prompt. [b] Paragraph-oriented extraction prompt. [c] Zilliz token-span output at threshold 0.3. [d] Native Provence setting with internal splitting for contexts longer than 512 tokens. [e] Threshold 0.2 with min-span length 10 and merge gap 20. [f] `verbatim-rag-modern-bert-v2`, threshold 0.2 with min-span length 30 and merge gap 20.

ModernBERT model also remains competitive on ACL gold (46.29 Word-F1), outperforming the public extractive baselines despite not being specialized only for the ACL Anthology. Our ACL-specialized model also achieves the highest word-level precision. Unlike the LLM extractors, it often abstains on irrelevant chunks. LLMs, in particular those used with paragraph-oriented prompts, achieve higher recall and higher span-level coverage, but achieve substantially lower precision, extracting evidence from many chunks that are irrelevant for the query. This trade-off is of particular importance in our context of retrieval-augmented question answering, where high-precision extraction models are effective filters of irrelevant search results. This difference is directly observable if we compare the results of our best model with an LLM-based extractor that achieves much higher recall. On the 100 chunks in the evaluation dataset, 53 of which had no gold spans annotated, our model chose not to predict any spans for 60 chunks while the paragraph-based Mistral model abstained only 35 times. We also illustrate this behavior with a cherry-picked example. For the query *hate speech detection downsampled training examples number*, one of the top retrieved chunks is a subsection describing the experimental dataset in the paper *EDAL: Entropy based Dynamic Attention Loss for HateSpeech Classification* (Fahim et al., 2023). This text provides lots of detail about topics closely related to the query, including statistics on class labels, but it is not at all concerned with downsampling. Our model correctly chose not to extract any spans, and so did the Zilliz highlighting model and some of the LLM-based extractors, including the default Mistral model, both Nemotron models, and the paragraph-based Qwen model. However, the four remaining models each choose some false positive spans, including texts on merging labels and texts as well as tables on dataset sizes.

6 Conclusion

We described an application of the VerbatimRAG architecture to over 100K research papers in the ACL Anthology, contributed a manually annotated dataset for the core extraction task, and presented a set of experiments showing that small customized encoder-decoder architecture trained with synthetic data outperforms zero-shot LLM-based extraction on this task, at a fraction of the cost. We release all components of our pipeline as open-source software. We believe that combining the VerbatimRAG approach with the task-oriented training of extractive models provides a blueprint for the efficient deployment of high-performing hallucination-free question answering systems across a variety of domains.

Limitations

The validity of our conclusions is limited by the size of the manually annotated dataset that was the basis of both quantitative and qualitative evaluation. The high complexity of the annotation task, described in detail in Section 3.2, also limited our ability to measure agreement between multiple annotators. to implement a rigorous adjudication process for resolving differences among annotators, or to develop detailed and objective annotation guidelines. We believe that all these steps will be possible if our approach is applied to more narrowly defined question answering use-cases that in turn lead to more objective extraction tasks. Furthermore, the extraction models trained using synthetic training data may reproduce unintended bias present in the output of LLMs, which may lead to such bias being reinforced and propagated by our models.

References

Parul Awasthy, Aashka Trivedi, Yulong Li, Meet Doshi, Riyaz Bhat, Vignesh P, Vishwajeet Kumar, Yushu Yang, Bhavani Iyer, Abraham Daniels, Rudra Murthy, Ken Barker, Martin Franz, Madison Lee, Todd Ward, Salim Roukos, David Cox, Luis Lastras, Jaydeep Sen, and Radu Florian. 2025. Granite embedding r2 models. *Preprint*, arXiv:2508.21085.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Masha Belyi, Robert Friel, Shuai Shao, and Atindriyo Sanyal. 2025. Luna: A lightweight evaluation model to catch language model hallucinations with high accuracy and low cost. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 398–409, Abu Dhabi, UAE. Association for Computational Linguistics.

Marcel Bollmann, Nathan Schneider, Arne Köhn, and Matt Post. 2023. Two decades of the ACL Anthology: Development, impact, and open challenges. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 83–94, Singapore. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen Mckeown. 2024b. Do models explain themselves? Counterfactual simulatability of natural language explanations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 7880–7904. PMLR.

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. 2025. Provence: efficient and robust context pruning for retrieval-augmented generation. *Preprint*, arXiv:2501.16214.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Md Fahim, Dr. Amin Ahsan Ali, Md Ashraful Amin, and Akm Mahbubur Rahman. 2023. EDAL: Entropy based dynamic attention loss for HateSpeech classification. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 775–785, Hong Kong, China. Association for Computational Linguistics.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2025. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *Preprint*, arXiv:2407.11005.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *Preprint*, arXiv:2310.18344.

Daniel Gildea, Min-Yen Kan, Nitin Madnani, Christoph Teichmann, and Martín Villalba. 2018. The ACL Anthology: Current state and future directions. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 23–28, Melbourne, Australia. Association for Computational Linguistics.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *Preprint*, arXiv:2410.12837.

Mark Hepple and Glyn Morrill. 1989. Parsing and derivational equivalence. In *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, England. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1077–1086, Uppsala, Sweden. Association for Computational Linguistics.

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

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *Preprint*, arXiv:2307.10169.

Adam Kovacs, Paul Schmitt, and Gabor Recski. 2025. KR labs at ArchEHR-QA 2025: A verbatim approach for evidence-based question answering. In *Proceedings of the 24th Workshop on Biomedical Language Processing (Shared Tasks)*, pages 69–74, Vienna, Austria. Association for Computational Linguistics.

Junyong Lin, Lu Dai, Ruiqian Han, Yijie Sui, Ruilin Wang, Xingliang Sun, Qinglin Wu, Min Feng, Hao Liu, and Hui Xiong. 2025. Scirgen: Synthesize realistic and large-scale rag dataset for scientific research. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 5619–5630, New York, NY, USA. Association for Computing Machinery.

Jennifer M. Logg, Julia A. Minson, and Don A. Moore. 2019. Algorithm appreciation: People prefer algorithmic to human judgment. *Organizational Behavior and Human Decision Processes*, 151:90–103.

Alejandro Lozano, Scott L Fleming, Chia-Chun Chiang, and Nigam Shah. 2023. Clinfo.ai: An open-source retrieval-augmented large language model system for answering medical questions using scientific literature. *Preprint*, arXiv:2310.16146.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 295–337, Bangkok, Thailand. Association for Computational Linguistics.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. 2024. Hallucination-free? assessing the reliability of leading ai legal research tools. *Preprint*, arXiv:2405.20362.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Saif M. Mohammad. 2020. NLP scholar: A dataset for examining the state of NLP research. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 868–877, Marseille, France. European Language Resources Association.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand. Association for Computational Linguistics.

Monarch Parmar, Naman Jain, Pranjali Jain, P. Jayakrishna Sahit, Soham Pachpande, Shruti Singh, and Mayank Singh. 2020. Nlpexplorer: Exploring the universe of nlp papers. In *Advances in Information Retrieval*, pages 476–480, Cham. Springer International Publishing.

Mark Perlin. 1991. LR recursive transition networks for Earley and Tomita parsing. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 98–105, Berkeley, California, USA. Association for Computational Linguistics.

Vipula Rawte, S. M Towhidul Islam Tonmoy, Krishnav Rajbangshi, Shravani Nag, Aman Chadha, Amit P. Sheth, and Amitava Das. 2024. Factoid: Factual entailment for hallucination detection. *Preprint*, arXiv:2403.19113.

Shaurya Rohatgi, Yanxia Qin, Benjamin Aw, Niranjana Unnithan, and Min-Yen Kan. 2023. The ACL OCL corpus: Advancing open science in computational linguistics. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10348–10361, Singapore. Association for Computational Linguistics.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. Ares: An automated evaluation framework for retrieval-augmented generation systems. *Preprint*, arXiv:2311.09476.

Juntong Song, Xingguang Wang, Juno Zhu, Yuanhao Wu, Xuxin Cheng, Randy Zhong, and Cheng Niu. 2024. RAG-HAT: A hallucination-aware tuning pipeline for LLM in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1548–1558, Miami, Florida, US. Association for Computational Linguistics.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *Preprint*, arXiv:2211.09085.

Felipe Viegas, Washington Cunha, Christian Gomes, Antônio Pereira, Leonardo Rocha, and Marcos Goncalves. 2020. CluHTM - semantic hierarchical topic modeling based on CluWords. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8138–8150, Online. Association for Computational Linguistics.

Vijay Viswanathan, Luyu Gao, Tongshuang Wu, Pengfei Liu, and Graham Neubig. 2023. DataFinder: Scientific dataset recommendation from natural language descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10288–10303, Toronto, Canada. Association for Computational Linguistics.

Cheney Zhang and Jiang Chen. 2026. How we built a semantic highlight model to save token cost for rag. https://huggingface.co/blog/zilliz/zilliz-semantic-highlight-model. Hugging Face community article, published January 15, 2026.

Ilana Zimmerman, Jadin Tredup, Ethan Selfridge, and Joseph Bradley. 2024. Two-tiered encoder-based hallucination detection for retrieval-augmented generation in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 8–22, Miami, Florida, US. Association for Computational Linguistics.

Ádám Kovács. 2026. Squeez: Task-conditioned tool-output pruning for coding agents. *Preprint*, arXiv:2604.04979.

Ádám Kovács and Gábor Recski. 2025. Lettucedetect: A hallucination detection framework for rag applications. *Preprint*, arXiv:2502.17125.

A Query generation example

B Detailed model comparison

Table 2 reports containment and coverage metrics for the extractor configurations evaluated in Table 1. All metrics are computed on the full 100-row benchmark, including the 53 irrelevant query–chunk pairs as negative examples.

C Threshold selection for the student model

The student is a binary token classifier, so span decisions depend on a probability threshold at inference. Table 3 reports the all-row gold benchmark scores for the GTE-reranker student at $t \in \{0.2, 0.3, 0.4, 0.5\}$ with the same post-processing held constant: spans shorter than 10 characters are dropped, and neighbouring spans separated by at most 20 characters are merged.

The best Word-F1 is obtained at t=0.2. Increasing the threshold improves precision but reduces recall, which lowers F1 on this benchmark.

D Prompts for query generation

D.1 Question-type classification prompt

```
You are a researcher generating questions and
   answers to find relevant
information within a specific domain. Below
   are the potential question types.
Choose the type that best fits the field
   information and the user's purpose.

1. Verification: questions seeking a simple
   yes/no confirmation.
2. Disjunctive: questions presenting multiple
   alternatives.
3. Concept Completion: questions starting with
    Who/What/When/Where.
4. Example: questions asking for instances of
   a concept.
5. Feature Specification: questions about
   properties or characteristics.
6. Quantification: questions seeking numerical
    or measurable information.
7. Definition: questions asking for the
   meaning of a term or concept.
8. Comparison: questions asking for
   similarities or differences.
9. Interpretation: questions asking for
   inference over observed patterns.
10. Causal Antecedent: questions about causes
   or reasons.
11. Causal Consequence: questions about
   outcomes or results.
12. Goal Orientation: questions about
   objectives or intentions.
13. Instrumental/Procedural: questions asking
   how to achieve a goal.
14. Enablement: questions about conditions
   enabling an action.
15. Expectation: questions about anticipated
   or missing outcomes.
16. Judgmental: questions asking for
   evaluation or opinion.
17. Assertion: statements indicating lack of
   knowledge.
18. Request/Directive: requests to summarize,
   analyze, or search.

Task: Based on the following text from a
   research paper, return the most
appropriate 3 question types that could be
   answered by this text. Give me the
name of each type and not other information.
   Return ONLY valid JSON -- an
array of objects, no markdown or explanations.

Text: {chunk}
```

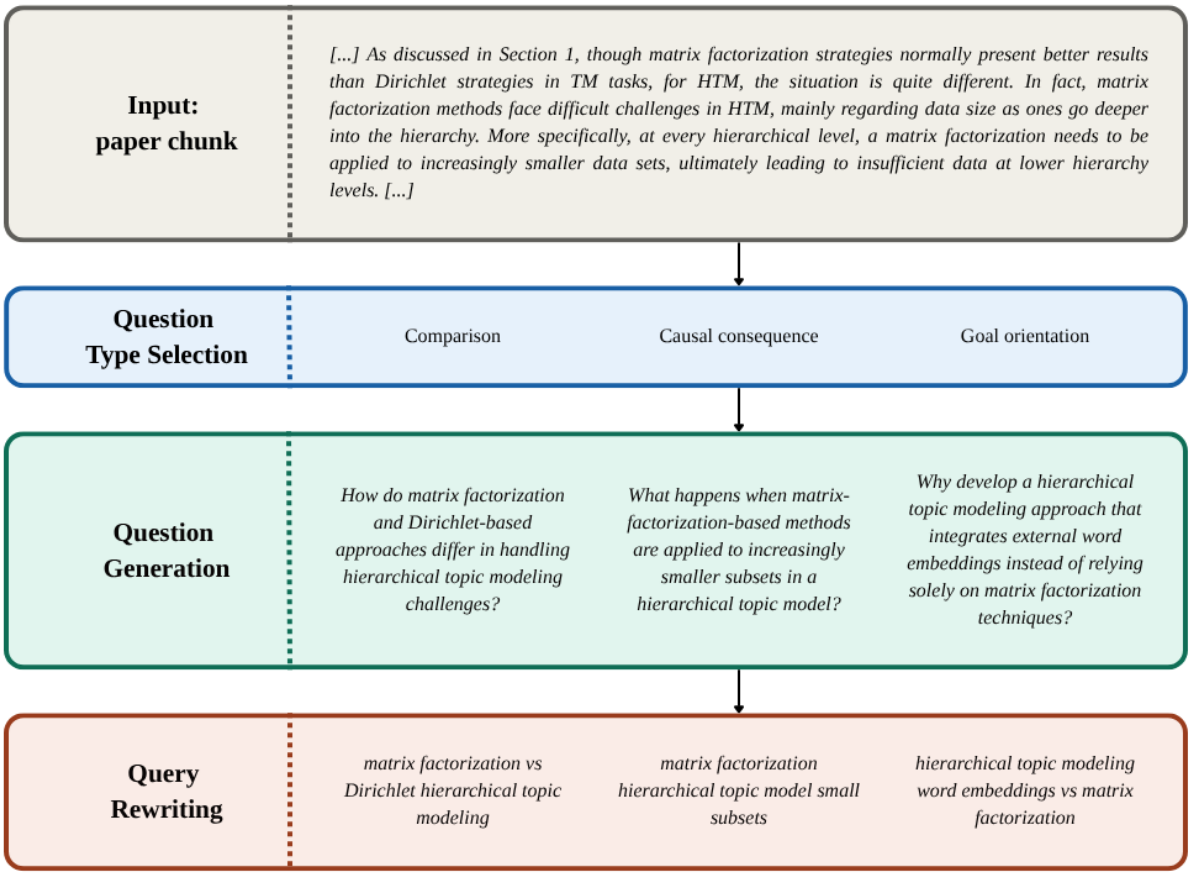


Figure 1: Generation of synthetic user queries, based on the ScIRGen methodology. The example shows a chunk from the paper *CluHTM - Semantic Hierarchical Topic Modeling based on CluWords* (Viegas et al., 2020)

D.2 Question generation prompt

```
You are a researcher asking questions aiming
   to find information in research
papers.

Content of paper: {chunk}

Please generate one question that can be
   answered by the above text and which
belongs to the question type below.

- Question Type: {q_type}
- Question Description: {q_def}
- Question Example: {q_ex}

Instructions:
1. Only return a question without any other
   information.
2. Use neutral terms like "a dataset", "data
   collection method", or
   "research approach", instead of references
   like "the study" or
   "this dataset".
3. The question should be short and simple,
   resembling what a user might
   type into a search engine.
4. The question should be answerable based on
   the text above.
```

D.3 Query rewriting prompt

```
You are a researcher using a search engine to
   find information.

Your question: {question}

Please generate a search query that you would
   use to find the answer to this
question.

Instructions:
1. Only return a search query without any
   other information.
2. The query should be short and simple,
   resembling what a user might type
   into a search engine.
3. The query does not need to be grammatical.
```

E Prompts for extraction

E.1 Default VerbatimRAG extraction prompt

```
Extract EXACT verbatim text spans from
   multiple documents that answer the
question.

Rules
```

	Containment @ t			Coverage @ t		
Model	0.5	0.8	1.0	0.5	0.8	1.0
Mistral Small 2603	0.497	0.462	0.439	0.577	0.397	0.179
Mistral Small 2603 + paragraph	0.390	0.336	0.277	0.756	0.654	0.487
Nemotron-120B-A12B	0.429	0.366	0.366	0.526	0.436	0.436
Nemotron-120B-A12B + paragraph	0.449	0.313	0.299	0.628	0.628	0.590
GLM-5	0.584	0.542	0.529	0.538	0.346	0.205
Qwen 3.6 35B	0.533	0.510	0.490	0.410	0.256	0.154
Qwen 3.6 35B + paragraph	0.478	0.370	0.348	0.615	0.513	0.449
Zilliz Semantic Highlight	0.395	0.355	0.333	0.192	0.090	0.026
Provence reranker-pruner	0.292	0.228	0.216	0.410	0.218	0.167
ACL-Verbatim GTE-reranker (t=0.2 + merge)	0.482	0.315	0.273	0.423	0.372	0.359

Table 2: Detailed extractor metrics on the full 100-row gold benchmark. Containment measures how much of a predicted span lies inside a gold span; coverage measures how much of a gold span is covered by a prediction.

Threshold	Word-P	Word-R	Word-F1	IoU-F1@0.5
0.2	0.654	0.454	**0.536**	**0.389**
0.3	0.667	0.421	0.516	0.338
0.4	0.678	0.403	0.506	0.365
0.5	**0.701**	0.380	0.493	0.336

Table 3: Threshold sweep for the GTE-reranker student on the full 100-row gold benchmark.

```
1. Extract only text that explicitly addresses
    the question.
2. Never paraphrase, modify, or add to the
   original text.
3. Preserve original wording, capitalization,
   and punctuation.
4. Order spans within each document by
   relevance, most relevant first.
5. Include complete sentences or paragraphs
   for context.

Output format
Return a JSON object mapping document IDs to
   span arrays ordered by relevance:
{
  "doc_0": ["most relevant span", "next most
   relevant span"],
  "doc_1": ["most relevant from doc 1"],
  "doc_2": []
}

If no relevant information exists in a
   document, use an empty array.

Your task
Question: {{ question }}

Documents:
{{ documents }}

Extract verbatim spans from each document:
```

E.2 Paragraph-style extraction prompt

```
Extract verbatim supporting passages from each
    document that answer the
question.

What to extract
A supporting passage is the complete portion
   of the document a researcher
would highlight to justify the answer,
   including:
- the sentence(s) that directly address the
   question;
- preceding setup sentence(s) that introduce
   the topic, methodology, or
  figure being referenced;
- concluding interpretation sentence(s) that
   summarize implications;
- table captions when the table itself is
   relevant.

Prefer a single continuous paragraph over
   multiple fragments of the same
paragraph. Only split into multiple spans when
    relevant content is in
non-adjacent parts of the document.

Rules
1. Use EXACT text from the document; no
   paraphrasing or edits.
2. Preserve original wording, capitalization,
   and punctuation.
3. If no passage in the document supports the
   answer, return an empty array.
4. Order spans within each document by
   relevance, most relevant first.

Output format
Return JSON mapping document IDs to arrays:
{
  "doc_0": ["first supporting passage", "
   second supporting passage"],
  "doc_1": ["passage from doc 1"],
  "doc_2": []
}

Your task
Question: {{ question }}

Documents:
{{ documents }}

Extract supporting passages from each document:
```