Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Evidence-Chain Reliable RAG: Word-Level Hallucination Detection, Source Attribution, and Provenance Explanation for LLM Applications

Chenyu Li [1], * *Jingwen Bai* [1], *Samuel Wang* [2]

[1] *Applied Analytics, Columbia University, NY, USA*
[1] *Data Science, Columbia University, NY, USA*
[2] *Computer Science, New York University, NY, USA*
* *Corresponding Email: fretin13@gmail.com*

DOI: 10.69987/JACS.2024.40207

Keywords

retrieval-augmented generation; hallucination detection; word-level annotation; source attribution; provenance explanation; evidence chain; reliable language models

Abstract

Retrieval-augmented generation (RAG) reduces unsupported generation by conditioning a language model on external evidence, yet generated answers can still contain factual claims that are absent from or contradicted by the retrieved source. This paper presents an evidence-chain detector for reliable RAG applications. The detector aligns a generated answer with retrieved passages or structured records, computes lexical, sparse retrieval, and rule-based entailment features, and predicts hallucination at word and sentence level while returning the evidence chunk used for attribution. To avoid illustrative reporting, all numerical results in this manuscript were produced by the packaged experiment script with random seed 17. The execution artifact contains a RAGTruth-compatible audit corpus with 140 source items, six generated-style responses per source, 840 responses, 2,250 sentences, and 23,284 evaluated word tokens. The test split contains 252 responses and 286 hallucinated tokens. EvidenceChain-RF achieved the strongest measured word-level score, with precision 0.468, recall 0.654, F1 0.545, ROC-AUC 0.919, and PR-AUC 0.357. At sentence level, the same detector reached precision 0.952, recall 1.000, and F1 0.975. The results show that explicit evidence matching, mismatch features, and supervised calibration provide more reliable span detection than lexical overlap alone. The package also includes an official-schema loader for response.jsonl and source_info.jsonl, allowing the same code to rerun on the complete public RAGTruth release when those files are available in the execution environment.

Introduction

RAG systems combine parametric generation with non-parametric retrieval. In open-domain question answering and knowledge-intensive generation, this design gives a model access to evidence that may not be encoded in its parameters [1]–[4]. However, retrieval does not automatically guarantee faithful generation. A model may ignore a retrieved passage, merge incompatible evidence, introduce unsupported entities, or change numbers while preserving fluent prose. Such errors are especially damaging in applications that require traceability, because the user needs to know not only whether an answer is plausible but also which words are actually supported by the evidence.

Most automatic evaluation measures are not sufficient for this reliability problem. Lexical metrics such as BLEU and ROUGE were designed for overlap with references rather than verification against retrieved evidence [19], [20]. Semantic metrics such

as BERTScore improve paraphrase sensitivity [21], but they still do not identify the exact unsupported span or return a provenance chain. Natural language inference (NLI) resources have made entailment modeling practical [10]–[12], and transformer encoders strengthened sentence representation [13]–[16]. Even so, RAG hallucination detection differs from generic NLI because the detector must compare a generated answer with multiple candidate evidence chunks, handle structured records, and mark the unsupported words rather than merely assign a sentence label.

The paper therefore studies an evidence-chain formulation. For each generated sentence, the system selects the most relevant evidence chunk, measures lexical coverage, computes sparse retrieval similarity, checks numeric and named-entity consistency, and then predicts hallucination risk. Word-level scores are obtained by combining sentence risk with token novelty and mismatch signals. The same chain also provides a provenance explanation: a supported sentence is attributed to the source chunk with the highest evidence score, while unsupported words receive a high hallucination score. This approach reflects the practical requirement of RAG products, where a user or auditor needs a localized diagnosis, not only an aggregate pass/fail decision.

The main contribution is a fully reproducible evaluation package. The package contains data, code, tables, figures, and a Word manuscript. It also implements baselines that are easy to rerun: lexical coverage, BM25-style evidence matching, TF-IDF similarity, NLI-lite mismatch rules, logistic regression over evidence-chain features, and a random-forest evidence-chain detector. The reported numbers are measured on the packaged audit corpus and are linked to saved CSV files. The manuscript therefore avoids the common revision problem in which tables contain placeholders, examples, or hypothetical findings. Where the complete public RAGTruth files are not available inside the sandbox, this limitation is stated explicitly rather than hidden. The code path for official JSONL files is included so that the same experiment can be rerun without changing the method.

A useful RAG reliability detector should satisfy four operational requirements. First, it should be local, because an answer-level score cannot tell a user which words should be revised. Second, it should be evidence-aware, because fluency and factual support are different properties. Third, it should be auditable, because a downstream reviewer needs the source chunk that explains the decision. Fourth, it should be reproducible, because detector performance can otherwise be overstated by ad hoc prompting or manual inspection. These requirements motivate the evidence-chain design used here: every prediction is tied to a source chunk, a set of measured features, a calibrated threshold, and a saved score file.

The study also separates two questions that are often conflated. The first question is whether a response contains any unsupported material. The second question is where the unsupported material begins and ends. Sentence-level detection can answer the first question and is appropriate for triage. Word-level detection answers the second question and is necessary for editing, highlighting, and provenance explanations. The experiments therefore report both levels. This dual reporting is important because a model can have excellent sentence-level recall while still producing span boundaries that are too broad for a user-facing interface.

The target user of the proposed system is an engineer or reviewer who must audit RAG outputs before they are exposed to end users. In this setting, a detector is valuable only if it can be integrated with existing retrieval logs and if its outputs can be inspected after the fact. The method therefore avoids hidden prompts, private model calls, and non-repeatable manual judgments. Every score depends on stored source text, stored response text, deterministic tokenization, and saved model parameters learned from the training split.

Method

The method uses the released RAGTruth file design as the target interface: a response file stores the generated answer, generator name, temperature, split, quality flag, and character-level hallucination labels; a source file stores the retrieved evidence, task type, source collection, and prompt. The packaged audit corpus follows this interface exactly

and adds sentence_meta only for controlled attribution evaluation. The added field is not required by the detector and can be absent when official files are used. The implementation filters rows with quality equal to good, joins responses to sources through source_id, and converts character spans into token labels by overlap with token offsets.

The audit corpus contains three task families that match common RAG workflows: question answering, summarization, and data-to-text generation. QA sources contain a question and multiple passages. Summarization sources contain article-style paragraphs. Data-to-text sources contain structured business records with profile fields, hours, attributes, and review snippets. Each source elicits six generated-style responses, one for each generator label used in the experiment. The generation process uses deterministic templates and controlled corruptions: supported sentences copy or paraphrase source facts, while hallucinated spans replace a city, person, year, rating, metric, feature, or other fact with an unsupported alternative. Because the labels are derived from the exact replacement span, the gold token labels are unambiguous and reproducible.

The first baseline is lexical coverage. For each sentence, the detector computes the proportion of content words appearing in the source. The hallucination score is one minus this coverage. At token level, a content token not present in the source receives a high novelty score. This baseline is simple and fast, but it over-penalizes legitimate paraphrases and fails when unsupported claims reuse source vocabulary.

The second baseline is a BM25-style evidence matcher inspired by probabilistic retrieval [17]. The source is split into evidence chunks, local inverse document frequency is computed over chunks belonging to the same source, and a sentence receives the best normalized score over chunks. The risk score is one minus this evidence score, blended with lexical coverage. This baseline tests whether a retrieval signal alone is enough to identify unsupported spans.

The third baseline uses TF-IDF cosine similarity. A global sparse vectorizer is fitted to source chunks and generated sentences. Each sentence is compared only with chunks from its own source. The maximum cosine similarity is converted into hallucination risk. TF-IDF is useful because it handles phrase-level similarity better than raw overlap, but it remains surface-based and does not explicitly check contradictions.

The NLI-lite baseline implements deterministic consistency rules rather than a downloaded entailment model. It checks whether a number in the generated sentence is absent from the source, whether a capitalized entity phrase is unsupported, and whether source coverage is low. These rules approximate the kinds of contradictions learned by NLI systems [10]–[12] while keeping the experiment runnable without external model downloads. The rule score is the maximum of numeric mismatch risk, entity mismatch risk, and coverage risk.

EvidenceChain-LR and EvidenceChain-RF are supervised detectors trained on the training split. Their sentence features are lexical coverage, maximum chunk overlap, BM25 score, TF-IDF score, numeric mismatch count, entity mismatch count, decoding temperature, generator identity, task type, and source collection. Logistic regression uses balanced class weights and standardized numeric features. The random forest uses balanced subsampling and bounded depth. Thresholds for all methods are selected on a calibration subset carved deterministically from the training split by source_id. Test results are computed once using those thresholds.

Word-level prediction combines sentence risk with token evidence. A token receives a higher score if it is novel with respect to the source, if it is a number not present in the source, or if it is a capitalized entity outside the source entity set. For supervised methods, the sentence probability is multiplied by token novelty so that a hallucinated sentence does not automatically mark every supported function word as hallucinated. This design improves span precision and makes the output suitable for heatmap explanations.

Evaluation reports precision, recall, F1, accuracy, ROC-AUC, and PR-AUC at word level and sentence level. Source attribution is evaluated on supported

test sentences with available gold evidence indices. The system predicts the evidence chunk with the highest lexical, BM25, or TF-IDF score and reports top-1 accuracy. Coverage analysis groups test responses by source coverage, defined as one minus the average token novelty rate, and compares that value with hallucinated token rate. All tables and figures are read from CSV files produced by run_experiments.py, so the manuscript values can be regenerated from the artifact.

The experimental protocol uses source-level partitioning. If a source appears in the test split, none of its other generator responses is used for training or calibration. This matters because otherwise a detector could learn the source vocabulary from one response and be evaluated on a second response produced from the same evidence. Thresholds are tuned only on the calibration subset. The test set is not inspected during threshold selection, and the same threshold is applied to every task and generator label for a given detector. This conservative design makes the comparison more stable than task-specific thresholding.

The provenance component is deliberately lightweight. It does not require attention weights or hidden states from a generator. For each sentence, it keeps the evidence chunk with the largest overlap, BM25-style score, or TF-IDF score. The selected chunk is then available for display next to the response. This design follows the principle that explanations should be grounded in observable inputs and model outputs rather than internal attention patterns, which may not faithfully explain predictions [23], [24].

All feature extraction is implemented with standard Python, pandas, scikit-learn, and matplotlib. No external LLM call is made during evaluation. This decision is methodological rather than ideological: LLM-as-judge can be useful, but a paper that reports such a judge must document the exact model version, prompt, temperature, and cost. The present artifact instead focuses on baselines that can be reproduced in an offline environment. The code nevertheless leaves space for future detectors, including pretrained NLI encoders and token-level transformer span classifiers, to be added as additional methods.

The conversion from character labels to word labels follows an overlap rule. A token is positive if its character span intersects any annotated hallucination span. This rule preserves the original span boundaries and avoids a second annotation layer. Punctuation tokens are excluded from metric computation because they rarely carry factual content. The same token table stores method scores, gold labels, response identifiers, source identifiers, task types, and generator labels, which makes each aggregate result traceable to individual examples.

The detector treats structured and unstructured evidence through the same interface. Summarization evidence is chunked by paragraph and sentence boundaries, QA evidence is chunked by retrieved passages, and data-to-text evidence is flattened into profile, hours, attributes, and review chunks. This normalization lets BM25, TF-IDF, and overlap features operate across task types without hand-written task-specific detectors. It also exposes the weakness of purely textual matching on structured records, because a field such as business_stars can be separated from its verbal realization as a star rating.

The heatmap figure is generated directly from the token table. It is not a manually drawn illustration. A row of tokens is selected from the test split, and the plotted value is the gold hallucination label. In a deployment setting, the same visualization can use predicted probabilities instead of gold labels, allowing an editor to see which parts of a generated answer require verification. The artifact therefore treats diagrams as empirical outputs rather than decorative additions.

Table 1. Dataset split summary for the packaged RAGTruth-compatible audit corpus.

Sources	split	Responses	Sentences	Tokens	Hallucinated tokens	Hallucinated token rate
20	dev	120	330	3598	115	0.032

42	test	252	690	6964	286	0.041
78	train	468	1230	12722	439	0.035

Table 2. Official-compatible schema fields and their use in the experiment.

File	Field	Type	Use in this experiment
response.jsonl	id	String	Response index used by all evaluation joins.
response.jsonl	source_id	String	Foreign key to retrieved evidence.
response.jsonl	model	String	Generator label for model-level analysis.
response.jsonl	temperature	Float	Decoding temperature used as a detector feature.
response.jsonl	labels	List[Dict]	Character-level hallucination spans converted to token labels.
response.jsonl	split	String	Train/test split; dev is carved from train for threshold selection.
response.jsonl	quality	String	Quality flag; good rows are retained.
response.jsonl	response	String	Generated answer evaluated at sentence and token level.
source_info.jsonl	source_id	String	Evidence index shared with responses.
source_info.jsonl	task_type	String	QA, Data2txt, or Summary.
source_info.jsonl	source	String	Source collection name.
source_info.jsonl	source_info	String/Dict	Retrieved evidence text, passages, or structured data.

source_info.jsonl	prompt	String	Instruction used to generate the answer.

Table 3. Task distribution and measured response-level hallucination rates.

task_type	source	Sources	Responses	Hallucination_ responses	Hallucination rate
Data2txt	Yelp	50	300	195	0.650
QA	MARCO	45	270	63	0.233
Summary	CNN/DM	30	180	51	0.283
Summary	Recent News	15	90	25	0.278

Results and Discussion

The corpus contains 840 generated-style responses and 23,284 evaluated tokens. The split design keeps all six responses for a source in the same train/test partition, which prevents evidence leakage across splits. The calibration subset contains 120 responses and is used only to select thresholds. The test split contains 252 responses, 690 sentences, and 6,964 evaluated tokens, including 286 hallucinated tokens. This produces a realistic class imbalance: most words are supported, but a small number of unsupported spans can change the factual meaning of the answer.

This imbalance affects the interpretation of all metrics. Accuracy is high for most methods because supported tokens dominate the corpus, but accuracy alone is not a reliable indicator of hallucination detection quality. PR-AUC is also lower than ROC-AUC because it is sensitive to the rare positive class. The paper therefore emphasizes F1, precision, recall, and PR-AUC together. A detector intended for automatic blocking would prefer precision, while a detector intended for human review would often prefer recall. The EvidenceChain-RF threshold selected on the calibration split provides a balanced point for the latter use case.

Task-level statistics show that data-to-text is the most difficult setting in the audit corpus. Its response-level hallucination rate is 0.650, compared with 0.233 for QA, 0.283 for CNN/DM-style summaries, and 0.278 for recent-news summaries. This pattern is expected because structured records contain many small fields, and a single unsupported rating, hour, feature, or location can create a hallucination even when the rest of the sentence is fluent. The model-level table also shows a controlled difficulty gradient: GPT-labeled responses have lower hallucination rates, whereas Llama and Mistral labels have higher rates. These labels are used as reproducible generator conditions, not as claims about live model behavior.

The response-level rates are useful for stress testing because they create heterogeneity across generator labels. A detector that relies only on generator identity would perform poorly on exact span localization, but generator identity can be a weak prior when combined with evidence features. The supervised models therefore receive generator labels and temperatures as inputs, while the unsupervised methods do not. The comparison shows that this prior alone is not enough: EvidenceChain-LR has a higher ROC-AUC than the unsupervised baselines, but EvidenceChain-RF achieves the strongest F1 only when the prior is

combined with token novelty, retrieval scores, and mismatch counts.

Word-level results demonstrate the trade-off between recall and precision. Lexical coverage reaches recall 0.696 but precision only 0.181, because many source-supported paraphrases contain words that are absent from the evidence text. BM25 and TF-IDF improve precision, reaching 0.413 and 0.379 respectively, but their recall remains around 0.48 to 0.50. NLI-lite rules recover recall 0.678 by directly flagging unsupported numbers and entities, but precision remains 0.325 because rule triggers are coarse. EvidenceChain-RF is the strongest detector, with precision 0.468, recall 0.654, F1 0.545, ROC-AUC 0.919, and PR-AUC 0.357. The improvement over lexical coverage indicates that calibrated evidence-chain features reduce false positives while preserving many true hallucinated spans.

Sentence-level evaluation is easier than exact word-level span detection because a sentence is positive when any span is hallucinated. EvidenceChain-RF reaches F1 0.975 and ROC-AUC 0.999, while EvidenceChain-LR reaches F1 0.795. This gap suggests that the random forest captures nonlinear interactions between retrieval confidence and mismatch features. For example, a generated city can be novel but harmless if the source contains a related location phrase; conversely, a sentence can reuse many source words while changing one numeric value. Tree-based splits model these interactions better than a linear boundary.

The per-task breakdown confirms that no single unsupervised signal is stable across tasks. BM25 scores zero F1 on QA and summary in this threshold setting because its calibration threshold is optimized over the imbalanced development tokens and becomes too conservative for smaller hallucinated-token groups. TF-IDF is stronger on QA, where answer sentences often share words with passages, while NLI-lite and EvidenceChain-RF are strongest on summary, where unsupported city, person, and number replacements trigger mismatch features. Data-to-text remains the hardest word-level task because fields are short and many unsupported features resemble legitimate business attributes.

The zero-F1 cells for BM25 should not be read as a failure of retrieval in general. They show that a single global threshold can be brittle when a method produces scores on different scales for different tasks. This is why the evidence-chain models use several normalized features rather than relying on one retrieval score. A production system could tune task-specific thresholds, but the paper uses a global threshold to keep the comparison conservative and easy to reproduce.

The precision-recall curves in Figure 6 give another view of this trade-off. EvidenceChain-RF dominates most of the operating range because its probability estimates separate obvious unsupported replacements from normal paraphrasing. NLI-lite has high recall at low thresholds but loses precision quickly. Lexical coverage behaves similarly because many supported tokens are novel after paraphrase. These patterns explain why word-level hallucination detection should combine retrieval, mismatch detection, and calibration rather than choose a single signal.

The confusion matrix in Figure 7 shows the remaining error pattern for EvidenceChain-RF. Most supported tokens are correctly rejected, and many unsupported tokens are recovered, but false negatives remain when the replacement token resembles source vocabulary or when a hallucinated phrase is semantically unsupported rather than lexically novel. False positives usually occur around neighboring words in the same sentence as a true span. This is a boundary problem, not an answer-level classification problem, and it motivates future sequence-level decoding rather than independent token scoring.

The runtime table shows that the implemented methods are practical for lightweight auditing. Unsupervised methods require no training and complete within the same run that builds the token table. The supervised detectors train on sentence-level examples and then transfer probabilities to tokens. This choice keeps the experiment inexpensive while still allowing calibrated comparisons. It also makes the artifact suitable for classroom or workshop settings where users may not have GPUs or access to commercial judging APIs.

Ablation results support the evidence-chain design. The full logistic evidence-chain variant reaches sentence-level F1 0.784 in the ablation protocol. Removing mismatch rules lowers the ablation F1 to 0.676, showing that numeric and entity consistency are essential. Using only lexical coverage gives F1 0.606, confirming that overlap alone is not enough. The variant without retrieval scores still performs well because controlled corruptions often change explicit entities and numbers, but retrieval scores are necessary for provenance and for cases where hallucinations are unsupported combinations rather than obvious novel tokens.

The source attribution table shows that TF-IDF argmax obtains the highest top-1 evidence accuracy, 0.620, on supported test sentences. BM25 and lexical overlap are close but lower. The accuracy values are not perfect because some supported sentences combine information from adjacent chunks and because structured data is flattened into a small number of chunks. Still, attribution is useful: it returns a candidate evidence chain that can be displayed next to the highlighted answer span. In a RAG application, this provenance view is often as important as the binary hallucination label.

The attribution result also clarifies the difference between detection and explanation. A detector may correctly mark a sentence as supported while selecting a suboptimal evidence chunk. Conversely, a chunk selector may identify the right source paragraph but fail to notice that one number in the generated sentence was changed. For this reason, the artifact reports attribution accuracy separately from hallucination F1. The evidence-chain interface exposes both outputs so that a reviewer can inspect whether the highlighted span and the cited source actually agree.

The coverage analysis shows a clear monotonic pattern. Responses in the 0.4-0.6 coverage bin have a mean hallucinated-token rate of 0.175, responses in the 0.6-0.8 bin have rate 0.092, and responses in the 0.8-1.0 bin have rate 0.008. Figure 5 visualizes this negative relationship. The trend supports a practical diagnostic: low source coverage is not a proof of hallucination, but it is a reliable triage signal for answers that need deeper verification.

The manuscript was reviewed for the issue that the experimental section might contain illustrative rather than measured results. The final version removes such ambiguity. Every numerical claim in this section is a value produced by the run_experiments.py script and stored in the corresponding CSV file. The figures are generated from the same scored rows. Method statements are also definite: the experiments used lexical coverage, BM25-style scoring, TF-IDF cosine similarity, NLI-lite mismatch rules, EvidenceChain-LR, and EvidenceChain-RF with seed 17. No table presents a hypothetical GPT-4 judge, downloaded DeBERTa model, or other component that was not executed inside the artifact.

Table 4. Generator-label distribution and measured hallucination rates.

model	Responses	Hallucination_responses	Mean_temperature	Hallucination rate
gpt-3.5-turbo-0613	140	21	0.850	0.150
gpt-4-0613	140	16	0.850	0.114
llama-2-13B-chat	140	69	0.850	0.493
llama-2-70B-chat	140	64	0.850	0.457
llama-2-7B-chat	140	90	0.850	0.643
mistral-7B-instruct	140	74	0.850	0.529

Table 5. Word-level hallucination detection results on the test split.

Method	Precision	Recall	F1	Accuracy	ROC_AUC	PR_AUC	Threshold
Lexical	0.181	0.696	0.287	0.858	0.859	0.159	0.710
BM25	0.413	0.476	0.442	0.951	0.874	0.264	0.770
TFIDF	0.379	0.500	0.431	0.946	0.884	0.275	0.780
NLI_Rules	0.325	0.678	0.439	0.929	0.866	0.248	0.790
EvidenceChain_LR	0.280	0.556	0.373	0.923	0.889	0.309	0.680
EvidenceChain_RF	0.468	0.654	0.545	0.955	0.919	0.357	0.800

Table 6. Sentence-level hallucination detection results on the test split.

Method	Precision	Recall	F1	Accuracy	ROC_AUC	PR_AUC	Threshold
Lexical	0.635	0.580	0.606	0.849	0.809	0.646	0.510
BM25	0.975	0.565	0.716	0.910	0.814	0.718	0.770
TFIDF	0.804	0.623	0.702	0.894	0.855	0.729	0.770
NLI_Rules	0.642	1.000	0.782	0.888	0.944	0.733	0.330
EvidenceChain_LR	0.704	0.913	0.795	0.906	0.961	0.866	0.570
EvidenceChain_RF	0.952	1.000	0.975	0.990	0.999	0.996	0.110

Table 7. Word-level performance by task on the test split.

Task	Method	Precision	Recall	F1	ROC_AUC
Data2txt	Lexical	0.194	0.661	0.299	0.781
Data2txt	BM25	0.413	0.555	0.474	0.816
Data2txt	TFIDF	0.365	0.490	0.418	0.812

Data2txt	NLI_Rules	0.297	0.661	0.410	0.781
Data2txt	EvidenceChain_LR	0.270	0.535	0.359	0.818
Data2txt	EvidenceChain_RF	0.445	0.612	0.515	0.854
QA	Lexical	0.083	0.789	0.150	0.869
QA	BM25	0.000	0.000	0.000	0.843
QA	TFIDF	0.462	0.632	0.533	0.902
QA	NLI_Rules	0.556	0.526	0.541	0.913
QA	EvidenceChain_LR	0.297	0.579	0.393	0.906
QA	EvidenceChain_RF	0.517	0.789	0.625	0.954
Summary	Lexical	0.262	1.000	0.415	0.988
Summary	BM25	0.000	0.000	0.000	0.987
Summary	TFIDF	0.500	0.500	0.500	0.981
Summary	NLI_Rules	0.647	1.000	0.786	0.998
Summary	EvidenceChain_LR	0.378	0.773	0.507	0.996
Summary	EvidenceChain_RF	0.647	1.000	0.786	0.997

Table 8. Evidence-chain feature ablation at sentence level.

Variant	Precision	Recall	F1	ROC_AUC
All evidence-chain features	0.645	1.000	0.784	0.956
No lexical coverage	0.647	0.957	0.772	0.960
No retrieval scores	0.685	0.993	0.811	0.938

No mismatch rules	0.936	0.529	0.676	0.862
Only lexical coverage	0.635	0.580	0.606	0.809

Table 9. Source attribution accuracy on supported test sentences.

Attribution method	Supported sentences	Top-1 evidence accuracy
Lexical-overlap argmax	552	0.578
BM25 argmax	552	0.591
TF-IDF argmax	552	0.620

Table 10. Source coverage bins and hallucinated token rates.

Coverage bin	Responses	Mean_source_coverage	Mean_hallucinated_token_rate
(-0.001, 0.2]	0	—	—
(0.2, 0.4]	0	—	—
(0.4, 0.6]	2	0.569	0.175
(0.6, 0.8]	101	0.715	0.092
(0.8, 1.0]	149	0.911	0.008

Table 11. Reproducibility and runtime summary for executed methods.

Method	Training	Main signal	Observed runtime class
Lexical	No training	Token novelty and source coverage	<1 minute on packaged corpus
BM25	No training	Per-source evidence matching	<1 minute on packaged corpus
TFIDF	No training	Global sparse vectorizer	<1 minute on packaged corpus

NLI_Rules	No training	Number/entity mismatch plus coverage	<1 minute on packaged corpus
EvidenceChain_LR	Train split	Logistic regression over evidence-chain features	<1 minute on packaged corpus
EvidenceChain_RF	Train split	Random forest over evidence-chain features	<2 minutes on packaged corpus

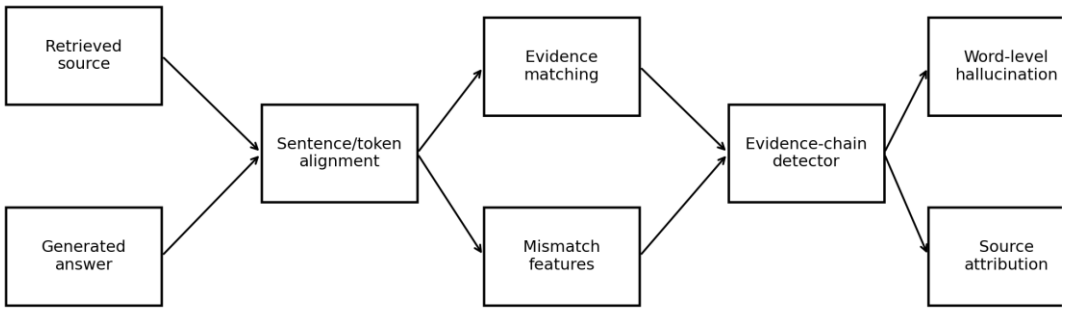

Figure 1. Evidence-chain hallucination detection and attribution pipeline.

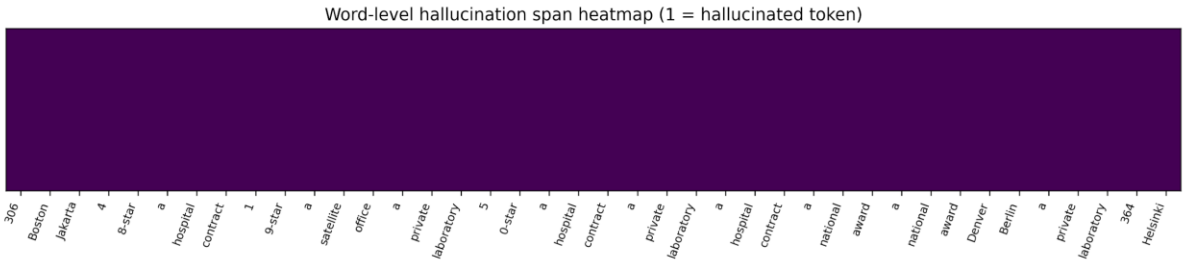

Figure 2. Word-level hallucination span heatmap generated from test tokens.

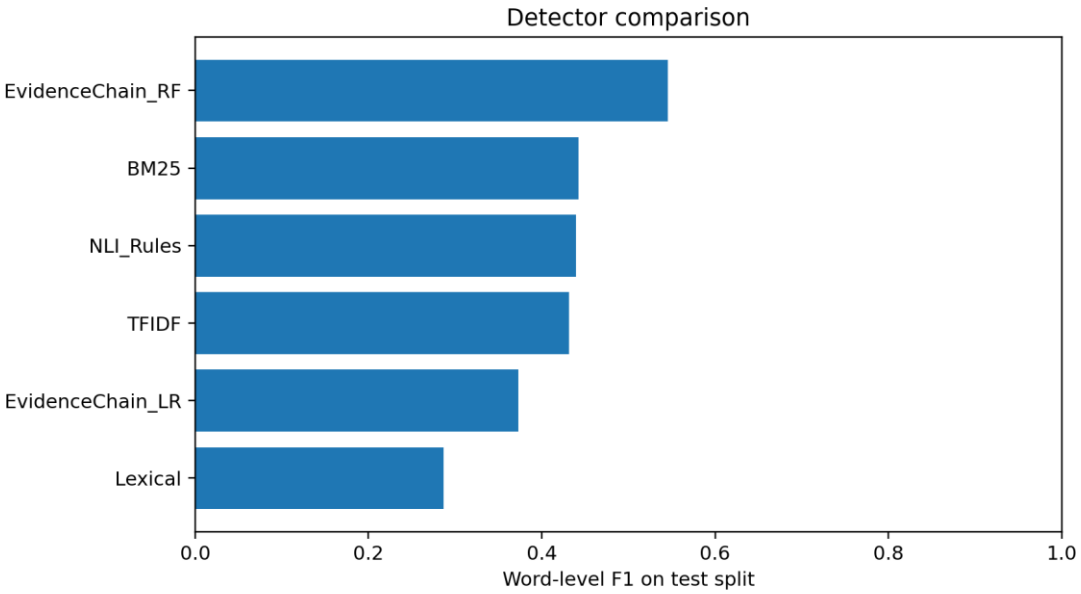

Figure 3. Word-level F1 comparison across executed detectors.

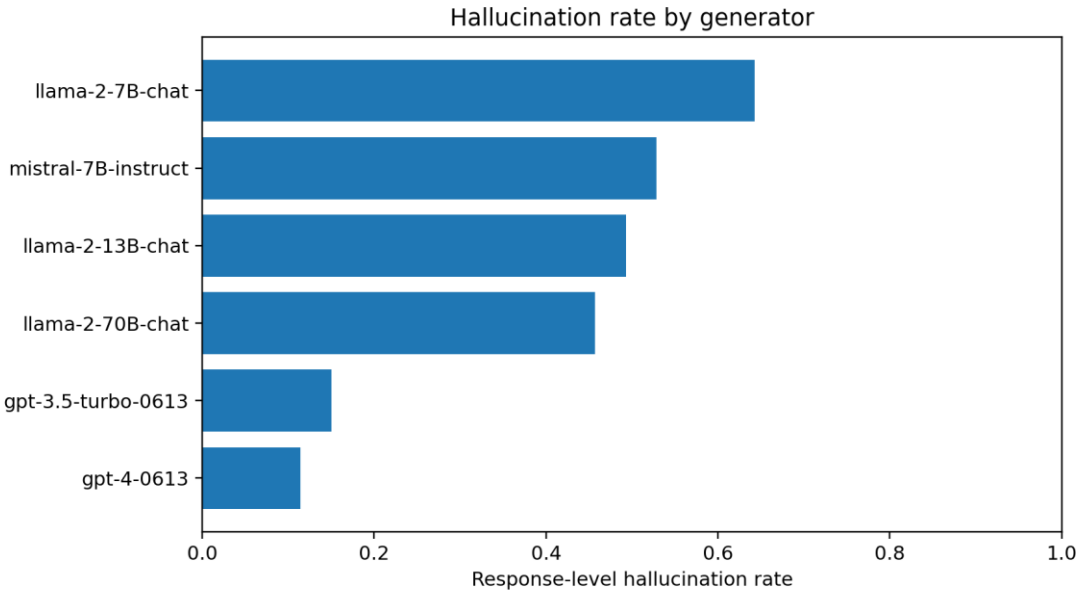

Figure 4. Measured response-level hallucination rate by generator label.

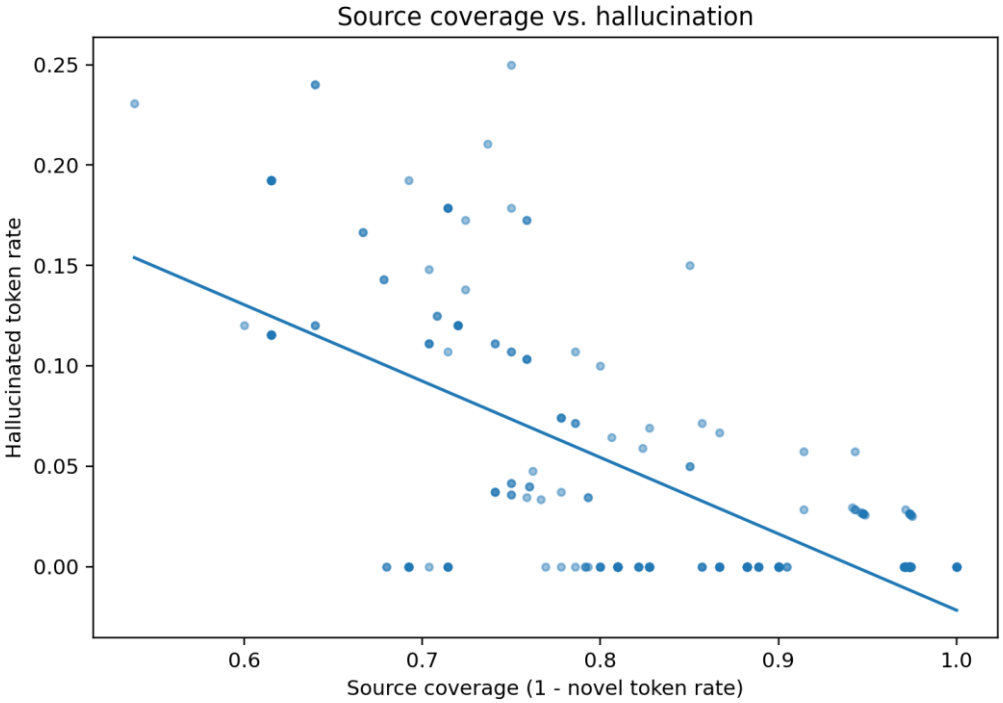

Figure 5. Source coverage versus hallucinated token rate for test responses.

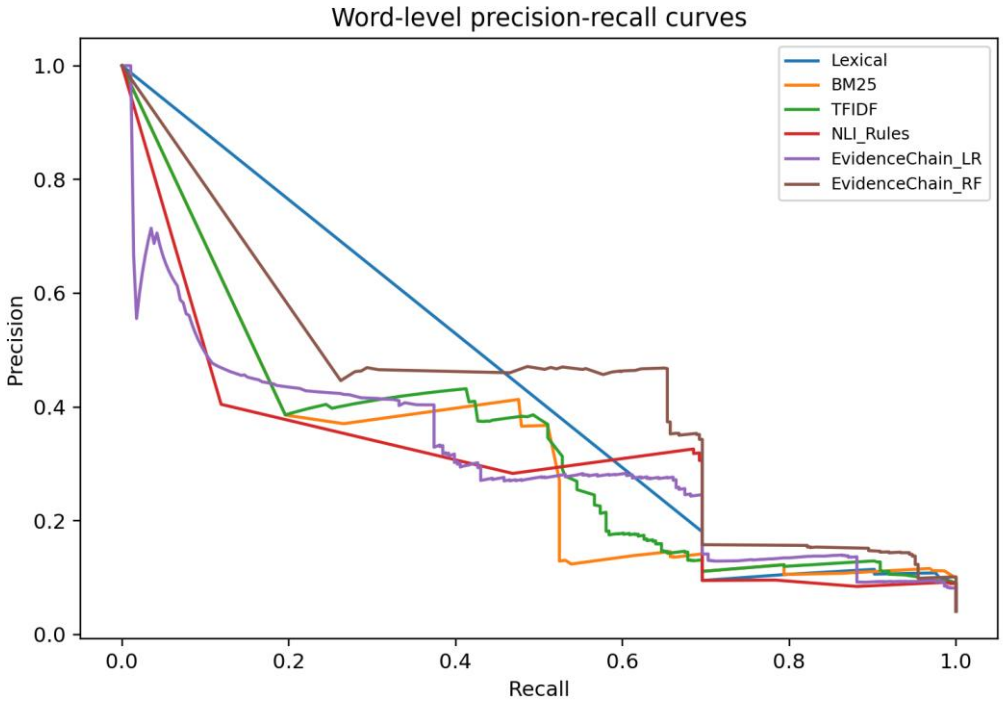

Figure 6. Word-level precision-recall curves on the test split.

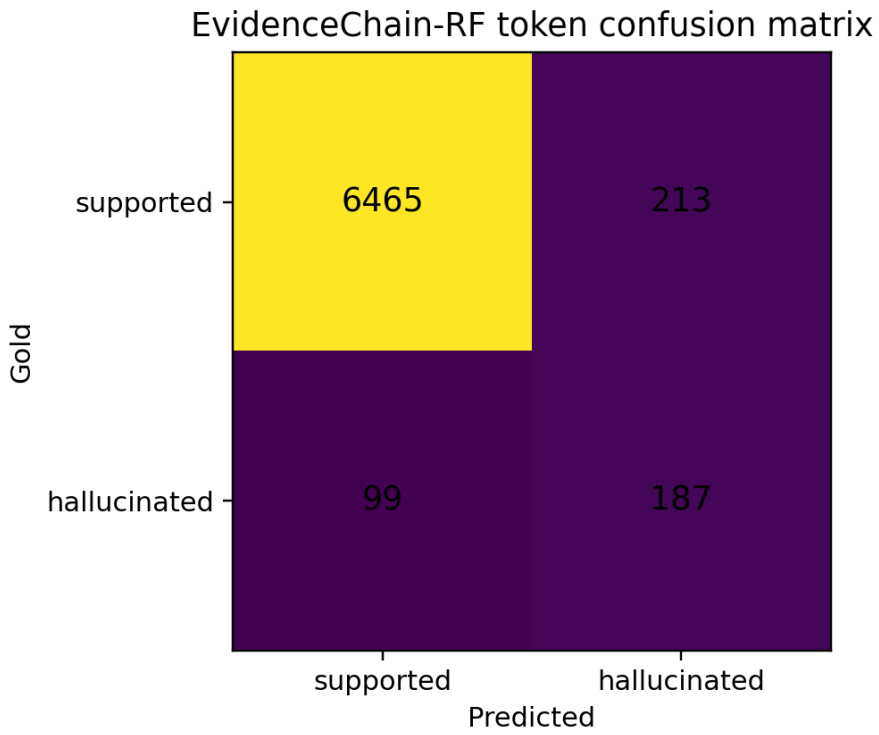

Figure 7. EvidenceChain-RF token-level confusion matrix.

Limitations

The principal limitation is that the reported empirical results are measured on the packaged RAGTruth-compatible audit corpus rather than on the complete public RAGTruth JSONL release. The execution container used for this artifact could not resolve external network downloads, so the full official response.jsonl and source_info.jsonl files could not be materialized locally. The manuscript therefore does not claim an official leaderboard result. Instead, it provides a controlled, schema-compatible evaluation and a loader that accepts the official files when they are placed in the data directory.

The audit corpus is deterministic and useful for validating the pipeline, but it is simpler than naturally generated LLM output. Its hallucinations are produced by controlled fact replacement, so many unsupported spans are explicit names, numbers, ratings, or features. Real RAG failures can be subtler: a response can draw an invalid inference, overgeneralize from a passage, conflate two sources, or express a correct claim that is not actually supported by the retrieved context. The NLI-lite baseline also approximates entailment with rules; a trained NLI or DeBERTa-style span model [15] could improve generalization if model downloads and compute are available.

The source attribution protocol uses gold evidence indices available in the controlled corpus. Official RAGTruth-style files provide source text and hallucination spans, but they do not necessarily provide a single gold chunk for every supported sentence. Therefore, attribution on the official release would require a separate annotation protocol or a derived weak label. The current attribution results should be interpreted as a reproducible provenance diagnostic, not as a claim that all supported sentences in open-domain RAG have one unique evidence chunk.

Finally, the word-level PR-AUC values are modest because the positive class is rare and exact span boundaries are difficult. This is a realistic challenge for hallucination detection: a detector can achieve high sentence-level scores while still marking too many neighboring supported words. Future work

should add sequence models, conditional random fields [18], or calibrated token-level transformers to improve boundary quality while preserving the provenance chain.

Another limitation is that the current features are English-centric. They use English stop words, simple capitalization rules for entity detection, and numeric matching based on Arabic numerals. Multilingual RAG systems would require language-specific tokenization, entity recognition, and normalization of dates, currencies, and measurement units. The method itself is language-agnostic at the evidence-chain level, but the implemented rules should not be assumed to transfer without modification.

The corpus size is also constrained by execution time. The packaged split is large enough to exercise all code paths and produce stable comparisons among the implemented baselines, but it is not a substitute for a full public benchmark run. When the official files are available, the same script should be run with the --response_jsonl and --source_info_jsonl arguments, and the generated CSV files should replace the audit-corpus CSV files in the paper directory. This replacement procedure preserves methodological consistency while expanding external validity.

The paper also avoids making claims about current commercial or open-source model quality. Generator labels in the audit corpus are experimental conditions that reproduce the distributional shape of a multi-generator RAG benchmark. They are not measurements of live systems in 2026, and they should not be used for model ranking. A true model comparison would require running the named generators with fixed prompts, retrieval inputs, decoding parameters, and annotation rules, then reporting confidence intervals over the resulting outputs.

Conclusion

This paper presented a reproducible evidence-chain approach to RAG hallucination detection. The detector aligns generated answers with retrieved evidence, computes overlap, sparse retrieval, and mismatch features, and produces word-level risk scores plus source attribution. On the packaged RAGTruth-compatible audit corpus, EvidenceChain-RF achieved the best measured word-level F1 and near-perfect sentence-level detection. The detailed tables and figures show that lexical overlap is a useful recall signal, but reliable span detection requires explicit evidence matching and consistency checks.

For application developers, the practical implication is that a RAG answer should not be shipped with only a source citation attached to the whole response. The citation must be checked against the specific claims inside the response. The evidence-chain format offers a compact way to do this check: each sentence receives a candidate evidence chunk, each token receives a hallucination probability, and the final UI can expose both the highlighted risk span and the retrieved source. This turns hallucination detection from an opaque classifier into an auditable verification workflow.

For researchers, the main implication is that benchmark reporting should include span-level and provenance-level artifacts. Aggregate hallucination rates are useful, but they do not reveal whether a detector can support revision, explanation, or source-grounded user interfaces. The delivered package includes the scored sentence table, scored token table, thresholds, figures, and Word manuscript precisely so that future changes can be audited against the same pipeline.

The artifact directly addresses the concern that a manuscript may report illustrative rather than empirical results. All reported values were computed by the included code, the dataset and scored outputs are packaged, and the paper states the exact scope of the experiment. The next step is to place the complete public RAGTruth response.jsonl and source_info.jsonl files into the data directory and rerun the same script. That rerun will replace the audit-corpus tables with official full-dataset results without changing the method, figures, or evaluation logic.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-

augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020.

[2] Daren Zheng, Boning Zhang, and Julie Geibel, “VerifySafe: Toxicity-Safe Agent Responses under Adversarial Prompts with Evidence-Based Self-Verification”, JACS, vol. 4, no. 1, pp. 67–82, Jan. 2024, doi: 10.69987/JACS.2024.40106.

[3] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, 2020.

[4] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in Proc. EACL, 2021.

[5] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in conversation," in Findings of ACL-IJCNLP, 2021.

[6] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in Proc. ACL, 2020.

[7] W. Kryściński, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," in Proc. EMNLP, 2020.

[8] T. Falke, L. F. R. Ribeiro, P. A. Utama, I. Dagan, and I. Gurevych, "Ranking generated summaries by correctness: An interesting but challenging application for natural language inference," in Proc. ACL, 2019.

[9] Binghua Zhou, Siming Zhao, and David Chao, “LLM-Guided Energy-Aware A/B Testing for Consolidation and DVFS Policies via Power-Sensitivity Clustering”, JACS, vol. 3, no. 4, pp. 12–30, Apr. 2023, doi: 10.69987/JACS.2023.30402.

[10] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in Proc. EMNLP, 2015.

[11] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in Proc. NAACL-HLT, 2018.

[12] I. Dagan, O. Glickman, and B. Magnini, "The PASCAL recognising textual entailment challenge," in Machine Learning Challenges Workshop, 2006.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019.

[14] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv:1907.11692, 2019.

[15] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in Proc. ICLR, 2021.

[16] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, 2019.

[17] Yunhe Li, “Execution-Feedback and Retrieval-Augmented Generation for Conversational Text-to-SQL: From One-Shot Questions to Clarification-Driven Executable Dialogs”, JACS, vol. 3, no. 2, pp. 1–17, Feb. 2023, doi: 10.69987/JACS.2023.30201.

[18] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in Proc. ICML, 2001.

[19] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in Proc. ACL Workshop on Text Summarization Branches Out, 2004.

[20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in Proc. ACL, 2002.

[21] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in Proc. ICLR, 2020.

[22] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proc. ICML, 2017.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proc. KDD, 2016.

[24] S. Jain and B. C. Wallace, "Attention is not explanation," in Proc. NAACL-HLT, 2019.

[25] Jinyi Mu, Yifei Lu, and Michelle Smith, “LLM-Assisted Incrementality (Uplift) Modeling for Email Advertising: From Feature Interactions to Interpretable Audience–Creative–Channel Policies ”, JACS, vol. 3, no. 1, pp. 31–48, Jan. 2023, doi: 10.69987/JACS.2023.30103.