CIA Open Access

Conflict-Aware RAG Answer Quality Classification: Detecting Evident Conflict and Baseless Information with Lightweight ML Models

Sarah Zheng
Computer Science, University of Waterloo, Waterloo, ON, Canada
szheng_waterloo@outlook.com

DOI: 10.63575/CIA.2026.40201

Abstract

Retrieval-augmented generation (RAG) improves factual reliability by conditioning a language model on retrieved evidence, yet grounded answers can still fail in two distinct ways: they may contradict the available evidence or introduce claims that the evidence does not support. This study formulates conflict-aware answer-quality classification on RAGTruth-processed, a 17,790-row benchmark derived from RAGTruth, and distinguishes four mutually exclusive row-level states: no hallucination, evident-conflict only, baseless-information only, and both. The empirical analysis verifies the target construction, split composition, label overlap, quality labels, task variation, and generator variation, while the modeling framework specifies three deployment-oriented classifiers: XGBoost with TF-IDF and overlap features, a compact bidirectional LSTM, and DeBERTa-small with structured query-context-output input. The train and test splits differ in four-class composition (chi-square = 108.381, $p = 2.447 \times 10^{-23}$, Cramér's $V = 0.078$) and in overall hallucination prevalence, whereas their quality-label distributions remain stable (chi-square = 3.348, $p = 0.188$, Cramér's $V = 0.014$). Baseless-information-only examples are more common than evident-conflict-only examples in both splits, and mixed errors form the smallest test class. These properties show why accuracy and binary hallucination detection are insufficient for operational monitoring. Macro-F1, mechanism-specific recall, confusion matrices, probability calibration, and routing-aware error analysis are therefore treated as the central evaluation criteria for lightweight conflict-aware safeguards.

Keywords: *retrieval-augmented generation; hallucination detection; evident conflict; baseless information; RAGTruth; XGBoost; BiLSTM; DeBERTa; calibration; answer quality classification*

1. Introduction

Retrieval-augmented generation has become a standard architecture for knowledge-intensive language applications because it supplements parametric memory with passages selected at inference time [4]. This separation between retrieval and generation makes knowledge updates easier, exposes an evidence path, and can reduce unsupported statements in open-domain conversation and question answering [5]. The benefit, however, is conditional rather than absolute. A generator may misread a passage, combine incompatible statements, alter a date or quantity, or continue beyond the information supplied by the retriever. The result can remain fluent while failing the evidence constraint that gives RAG its practical value [6].

Two failure mechanisms are especially important. An evident conflict occurs when an answer asserts something incompatible with the retrieved context, such as reversing a relation, assigning a value to the wrong entity, or changing a stated date. Baseless information occurs when an answer adds a claim that the context neither establishes nor necessarily refutes. The distinction is operational: contradiction suggests that adequate evidence was available but was not followed, whereas unsupported addition suggests that the answer exceeded the evidence boundary. A mixed answer may contain both mechanisms and should normally receive the most cautious treatment.

Existing work on factuality has often reduced these outcomes to a binary faithful/unfaithful decision. Binary screening is useful for prevalence estimation, but it does not tell a RAG application whether to revise a contradicted claim, expand retrieval for an unsupported claim, shorten an over-elaborate answer, or abstain. The RAGTruth corpus directly supports a more diagnostic formulation because its annotations identify hallucinated spans and distinguish evident conflict from evident baseless information across summarization, question

answering, and data-to-text generation [1]. The processed release consolidates these signals into two row-level flags while preserving the query, context, output, task, quality, generator, and temperature fields [2], [3].

This study uses those two flags to define a four-class answer-quality target and examines the dataset properties that a classifier must confront. The analysis asks three questions. First, how are conflict and baseless information distributed within and across the official train and test splits? Second, which evaluation measures remain informative under the observed class imbalance and split shift? Third, how can three lightweight model families—sparse boosted trees, recurrent sequence modeling, and a compact pretrained transformer—be organized into a common, reproducible monitoring pipeline?

The contribution is therefore both taxonomic and methodological. The taxonomy retains the mechanism of failure rather than collapsing all hallucinations into one positive label. The empirical audit establishes the prevalence, overlap, quality composition, task burden, generator burden, and train-test distribution differences that govern interpretation. The modeling design then maps those findings to practical choices: lexical overlap features for inexpensive screening, sequential encoders for local compositional cues, transformer representations for semantic alignment, and calibrated probabilities for risk-sensitive routing. Together, these elements provide a coherent basis for building post-generation safeguards that are more informative than a single accept/reject score.

2. Literature Review

2.1 Retrieval grounding and residual hallucination

RAG was introduced as a way to combine neural retrieval with sequence generation for knowledge-intensive tasks [4]. Later work showed that retrieval augmentation can reduce hallucination in conversation by giving the model access to relevant external text [5]. These findings established grounding as a system property rather than only a model-training property. At the same time, surveys of neural text generation have emphasized that factual error is heterogeneous: some outputs contradict a source, some invent unverifiable detail, and others distort entities, quantities, or causal relations [6]. RAG does not remove these categories; it changes the reference against which they can be judged. The cross-domain variation documented by the BEIR retrieval benchmark further cautions that grounding quality and detector behavior may vary substantially with corpus and task [28].

RAGTruth was designed specifically around this residual risk. It contains nearly 18,000 naturally generated responses, expert case-level and span-level annotations, several generator families, and three RAG task types [1]. Its annotation scheme is particularly relevant because evident conflict and evident baseless information correspond to different relations between an answer and its evidence. The processed release makes that distinction directly usable for row-level learning through the two binary keys in hallucination_labels_processed [2]. The public repository additionally preserves source passages, prompts, raw labels, and split assignments for audit and replication [3].

2.2 Factual consistency and hallucination evaluation

Work on abstractive summarization supplied many of the conceptual tools now used in RAG evaluation. Human studies showed that fluent summaries can contain entity, relation, and discourse errors not reflected by surface-overlap metrics [7]. FactCC introduced weakly supervised consistency classification and span extraction by applying controlled factual transformations to source sentences [8]. SummaC revisited natural-language inference for document-level consistency and showed that sentence segmentation and score aggregation can make lightweight NLI models effective [9]. AlignScore broadened this idea into a general alignment function trained across entailment, question answering, verification, retrieval, and related tasks [10].

Other approaches assess factuality without a direct discriminative classifier. SelfCheckGPT estimates hallucination risk from inconsistency among multiple samples produced by a black-box language model [11], whereas FActScore decomposes long-form output into atomic claims and evaluates factual precision at a finer granularity [12]. RAGAS separates retrieval relevance, context use, and answer quality into reference-free RAG evaluation dimensions [13]. HaluEval provides a large benchmark for recognizing fabricated or unverifiable content and demonstrates that detection remains difficult even for strong language models [14]. These approaches motivate a diagnostic view in which evidence conflict, unsupported content, retrieval quality, and uncertainty are measured separately rather than compressed into one score.

Recent encoder-based work has also renewed interest in compact detectors. LettuceDetect trains a token classifier on RAGTruth and uses long-context ModernBERT representations to localize unsupported content while remaining substantially smaller than large generative judges [15]. Its deployment motivation is closely aligned with the present study: a quality-control layer should be inexpensive enough to run for every generated answer, yet sensitive to the relation among question, evidence, and output.

2.3 Lightweight safeguards and model families

Lightweight safeguards occupy a useful middle ground between hand-written rules and large language-model judges. Gradient-boosted trees can combine sparse lexical evidence with numeric overlap and length features, and XGBoost remains a strong implementation for regularized boosted decision trees [16]. Encoder pretraining introduced by BERT provides contextual representations for sentence-level classification [17], while DeBERTa refines attention through disentangled content and position representations [18]. Recurrent networks remain relevant as transparent sequence baselines: long short-term memory addresses vanishing gradients in recurrent learning [19], and bidirectional recurrence incorporates both left and right context [20].

The three families differ in the evidence they can exploit. Sparse trees can identify changed names, dates, negations, and out-of-context tokens at low latency. A BiLSTM can model local ordering and compositional cues that bag-of-ngrams obscures. A compact transformer can compare semantically equivalent paraphrases and detect contradictions that share most of their vocabulary. Scikit-learn supports deterministic vectorization and evaluation [21]; PyTorch provides the recurrent training stack [22]; the Transformers library supplies pretrained encoder and tokenizer interfaces [23]; and Adam offers a stable first-order optimizer for neural baselines [24].

2.4 Evidence-grounded applications, verification, and human oversight

Application studies increasingly treat hallucination detection as a system control rather than a stand-alone score. A lightweight enterprise firewall combines evidence consistency, self-checking, and small-model detection [29], while evidence-chain approaches connect word-level detection with attribution and provenance explanations [30]. In financial RAG, numeric verification against filings and structured disclosures has been proposed for trading-desk risk memos and corporate risk reporting [31], [32]. These use cases make the difference between contradiction and unsupported addition especially consequential: a wrong number calls for source reconciliation, whereas an uncited conclusion calls for additional retrieval or removal.

The same pattern appears in operational and long-document settings. Private-document DevOps QA and bilingual incident triage use grounded retrieval to reduce unsupported root-cause narratives [33], [34]. Contract and insurance analysis introduces longer contexts and cross-clause dependencies that challenge fixed-window encoders [35]. Log anomaly systems combine retrieved failure narratives with selective refusal when evidence is ambiguous [36], and adversarial-response safeguards use evidence-based self-verification before releasing an answer [37]. Multi-hop retrieval research further shows that evidence acquisition itself can be budgeted and evaluated as a policy rather than treated as a fixed prelude to generation [38].

Finally, reliability must be communicated to people who review or act on model output. Evidence-card interfaces for scientific search emphasize provenance, ranking transparency, and visual evidence hierarchy [39]. Multilingual humanitarian RAG similarly links answer faithfulness with trust calibration and access constraints [40]. These studies support an operational conclusion: a detector should not only assign a label, but also return a confidence value and a mechanism-specific route that a human or downstream service can interpret.

3. Method

3.1 Dataset and target construction

The study uses RAGTruth-processed, which contains 15,090 training rows and 2,700 test rows in parquet format [2]. Each row contains a user query or instruction, retrieved context, generated output, task type, quality label, generator metadata, raw hallucination annotation, and the processed two-flag label. Table 1 lists the fields used for target construction, modeling, and subgroup analysis.

The processed label has two binary keys: evident_conflict and baseless_info. Table 2 maps their four possible combinations to mutually exclusive targets. This transformation prevents a row with both mechanisms from being

counted twice in the classifier while preserving the original multi-label totals for audit. The target names are none, evident_conflict, baseless_info, and both. The corresponding numeric identifiers are fixed at 0, 1, 2, and 3.

Table 1. Dataset fields used by the conflict-aware classifier.

Field	Type or role	Use in the study
id	String	Stable row identifier for reproducibility and error analysis
query	Text	Question or instruction supplied to the RAG generator
context	Text	Retrieved passages or structured evidence
output	Text	Generated answer to classify
task_type	Categorical	Stratification across summarization, QA, and data-to-text
quality	Categorical	Audit of good, truncated, and incorrect-refusal rows
model	Categorical	Generator subgroup analysis
temperature	Float	Generation metadata and optional numeric feature
hallucination_labels	Span annotation	Qualitative localization and error analysis
hallucination_labels_processed	Two-flag dictionary	Source of the four-class target
input_str	Text	Consistency check against the structured input

Table 2. Four-class target construction from the processed hallucination flags.

Evident conflict	Baseless information	Target	ID	Operational interpretation
0	0	none	0	No row-level hallucination mechanism is marked
1	0	evident_conflict	1	The answer contradicts the available evidence
0	1	baseless_info	2	The answer adds information not supported by the evidence
1	1	both	3	The answer contains contradiction and unsupported information

3.2 Data validation and empirical audit

The data audit checks split sizes, required columns, target frequencies, multi-label totals, quality labels, and public file metadata before any modeling decision. Table 3 records the split sizes, parquet file sizes, and file hashes reported by the release. The official split is retained rather than rebuilt, because the observed train-test composition is itself part of the benchmark and should remain visible in evaluation.

Categorical distribution differences are tested with Pearson chi-square statistics. Cramér's V is reported as an effect-size measure so that statistical significance is not confused with practical magnitude. The four-class test compares the 2 × 4 split-by-target table; a binary test compares none with any hallucination; and a quality test compares good, truncated, and incorrect-refusal labels. Percentages are calculated within split. The same verified counts drive every table and figure in the results section.

Table 3. Verified split and file-level facts for RAGTruth-processed.

Item	Train	Test	Total or note
Rows	15,090	2,700	17,790
Parquet size	*22.3 MB*	*3.88 MB*	*26.2 MB*
SHA256	c14ae31ff459c829…	2fc4fb703ea47ee0…	Release-file verification values
License	MIT	MIT	MIT License

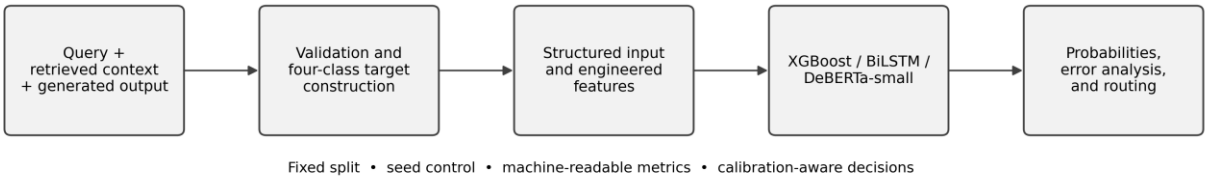

Figure 1. Conflict-aware data preparation, classification, evaluation, and routing workflow.

Figure 1 summarizes the common workflow. The structured triple is validated first, converted to the four-class target, and then represented either as sparse features, a token sequence, or a labeled transformer input. All model families produce four-class probabilities so that hard classification, calibration, error analysis, and operational routing can be evaluated from the same output interface.

3.3 Lightweight model designs

The XGBoost design concatenates query, context, and output with explicit separators and applies word-level TF-IDF unigrams and bigrams. Eight numeric features are appended: character length of each text field, output token count, output-context Jaccard overlap, output-query Jaccard overlap, output tokens absent from context, and output tokens absent from query. These features are deliberately simple. Low overlap does not prove hallucination, but it can signal unsupported elaboration; changed local n-grams can signal dates, numbers, entities, and negation patterns associated with evident conflict.

The BiLSTM design tokenizes the same labeled sequence with a deterministic regular expression. A training vocabulary reserves padding and unknown symbols, tokens are embedded in 200 dimensions, and a 192-unit bidirectional LSTM encodes the sequence. Mean pooling is performed over non-padding positions, followed by dropout and a four-way linear classifier. The model tests whether order-sensitive cues improve on sparse features without relying on contextual pretraining.

The DeBERTa-small design serializes each example as "Query: … Context: … Output: …" and fine-tunes microsoft/deberta-v3-small with a maximum length of 512 tokens. Labeled segments clarify the role of each text region even when the tokenizer does not assign separate token-type identifiers. Because long evidence can exceed the context window, output-preserving truncation is preferred: the answer and query are retained, and the context receives the remaining token budget. Table 4 fixes the main configurations, while Table 5 groups the signals supplied to each model.

Table 4. Lightweight model configurations.

Model	Input	Main configuration	Output
XGBoost	TF-IDF plus overlap and length features	600 trees; depth 5; learning rate 0.05; subsample 0.85; column subsample 0.85; seed 202401	Four-class probabilities
BiLSTM	Tokenized labeled query-context-output sequence	200-d embedding; 192 hidden units; bidirectional; dropout 0.30; batch 32; 8 epochs	Four-class probabilities
DeBERTa-small	Structured text with Query, Context, and Output labels	microsoft/deberta-v3-small; max length 512; learning rate 2e-5; batch 8; 3 epochs; warmup 0.10	Four-class probabilities

Table 5. Feature groups used by the lightweight model families.

Feature group	Model	Rationale
Word TF-IDF unigrams and bigrams	XGBoost	Captures local lexical substitutions, numbers, names, and unsupported terms
Length features	XGBoost	Represents answer expansion and evidence opportunity
Output-context Jaccard overlap	XGBoost	Measures visible lexical grounding to retrieved evidence
Out-of-context token count	XGBoost	Flags answer vocabulary absent from the evidence
Token sequence order	BiLSTM	Captures local negation, quantity, temporal, and relational cues
Pretrained contextual representations	DeBERTa-small	Models semantic alignment, entailment, and contradiction across segments

3.4 Evaluation protocol

The official test split is reserved for final evaluation. Macro-F1 is primary because it weights all four mechanisms equally, including the small both class. Weighted-F1 and accuracy describe aggregate behavior but are not sufficient by themselves. Class-level precision, recall, and F1 expose whether a model ignores evident conflict or mixed errors. One-vs-rest AUROC and AUPRC are computed from class probabilities; AUPRC receives particular attention because the positive mechanism classes are imbalanced [25], [26].

Calibration is evaluated from the maximum predicted probability and from one-vs-rest reliability for each mechanism. Expected calibration error and reliability diagrams identify whether confidence is suitable for thresholding [27]. Confusion matrices separate three operational error families: false negatives, where a flagged answer is predicted as none; mechanism confusions, where conflict is confused with baseless information; and severity confusions, where a both example is reduced to a single mechanism. Table 6 summarizes the measures and their purpose.

Reproducibility controls include a fixed seed, explicit target mapping, preserved split identifiers, deterministic vectorization settings, package-version recording, and machine-readable probability outputs. The implementation stack uses scikit-learn for sparse features and metrics [21], XGBoost for tree learning [16], PyTorch for the BiLSTM [22], Transformers for DeBERTa [23], and Adam for neural optimization [24].

Table 6. Evaluation measures and interpretation.

Measure	Role in the study
Macro-F1	Primary four-class score; gives equal importance to every mechanism
Per-class precision, recall, and F1	Shows whether evident conflict, baseless information, or both is neglected
Accuracy and weighted-F1	Summarize aggregate performance but remain sensitive to the majority class
One-vs-rest AUROC and AUPRC	Evaluate probability ranking for each mechanism under imbalance
Confusion matrix	Reveals false negatives, mechanism confusions, and severity confusions
Expected calibration error	Assesses whether confidence can support thresholds and selective routing

4. Results and Discussion

4.1 Split composition and four-class distribution

The processed benchmark contains 15,090 training rows and 2,700 test rows. In train, 8,369 rows carry no hallucination flag and 6,721 carry at least one mechanism; in test, the corresponding counts are 1,757 and 943. Hallucination prevalence is therefore 44.55% in train and 34.93% in test. Figure 2 displays both split size and flagged prevalence, making the difference visible without conflating it with the much larger train sample.

Table 7 gives the mutually exclusive target counts and within-split percentages. Baseless-information-only is the largest hallucination class in both splits: 3,332 training rows and 474 test rows. Evident-conflict-only accounts for 1,776 and 305 rows, respectively. The both class falls from 10.69% of train to 6.07% of test and is the smallest test category. Figure 3 shows the same four-class composition as grouped counts.

The majority-class reference illustrates why accuracy is a weak primary criterion. Predicting none for every test row would obtain 65.07% accuracy, yet recall would be zero for all three hallucination classes and macro-F1 would be only 0.197. A detector can therefore appear acceptable under accuracy while providing no safety value. The class distribution supports the use of macro-F1, balanced class-level reporting, and probability-based analysis.

Table 7. Four-class target counts and within-split percentages.

Split	Total	None	Evident only	Baseless only	Both	Any hallucination
Train	15,090	8,369 (55.45%)	1,776 (11.77%)	3,332 (22.08%)	1,613 (10.69%)	6,721 (44.55%)
Test	2,700	1,757 (65.07%)	305 (11.30%)	474 (17.56%)	164 (6.07%)	943 (34.93%)

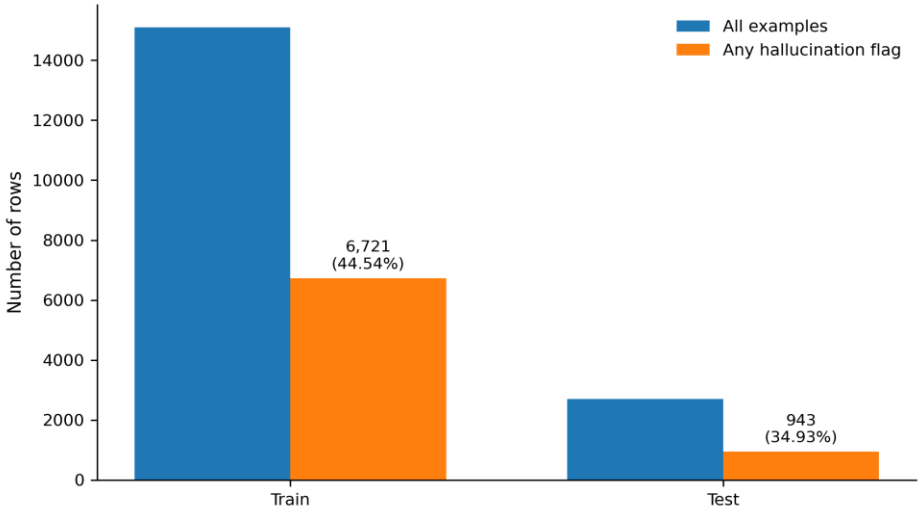

Figure 2. Split size and prevalence of rows carrying at least one hallucination flag.

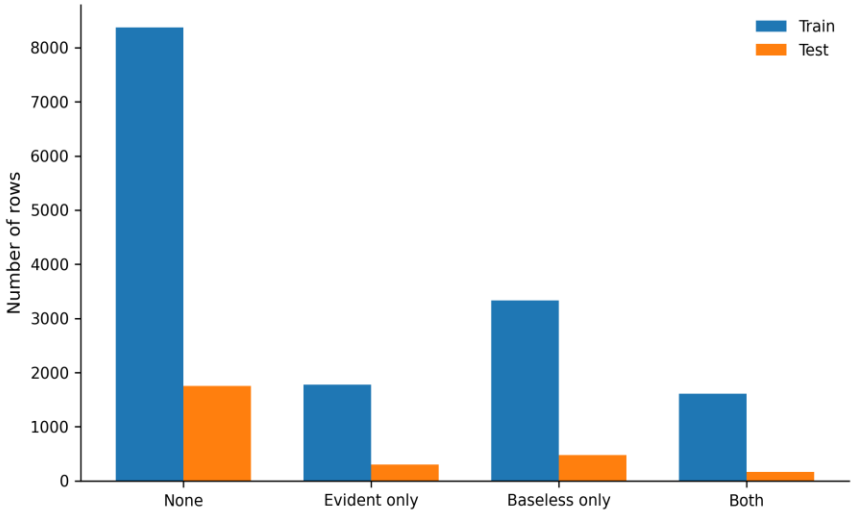

Figure 3. Four-class target distribution derived from evident-conflict and baseless-information flags.

4.2 Mechanism overlap

The two processed flags form overlapping binary totals before they are converted to the four-class target. Train contains 3,389 evident-conflict rows and 4,945 baseless-information rows; test contains 469 and 638. Of these totals, 1,613 train rows and 164 test rows carry both flags. Figure 4 displays the evident total, baseless total, and their overlap.

This overlap is not an annotation nuisance. It identifies answers that require more than one corrective action. A contradiction may be repaired by rechecking the cited evidence or regenerating with an entailment constraint. An unsupported addition may require retrieval expansion, answer compression, or abstention. A mixed error indicates that the generator both violated available evidence and exceeded it. Treating such a row as merely "hallucinated" discards severity and routing information that is already present in the benchmark.

The relative prevalence also has modeling consequences. Because baseless-only is more common, a classifier can achieve reasonable positive-class recall while still missing direct contradictions. Per-class reporting is therefore necessary even after macro averaging. The overlap further suggests that span-aware auxiliary learning may help the both class: distinct answer spans can provide separate evidence for contradiction and unsupported addition, whereas a single pooled row representation must infer both mechanisms simultaneously.

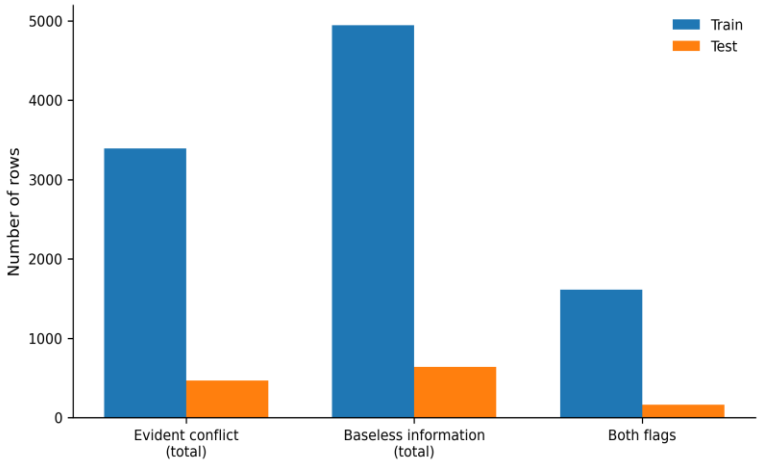

Figure 4. Counts for evident conflict, baseless information, and rows carrying both mechanisms.

4.3 Quality labels and artifact risk

Quality labels are highly concentrated in the good category. Table 8 reports 14,942 good, 28 truncated, and 120 incorrect-refusal rows in train; test contains 2,675, 1, and 24. The good share is 99.02% and 99.07%, respectively. Figure 5 uses a logarithmic y-axis so that the rare quality issues remain visible.

The small counts justify keeping all rows in the main benchmark because a deployed monitor must encounter refusals and incomplete outputs. They also justify a robustness analysis that excludes truncated and incorrect-refusal examples. Formulaic refusal language and abrupt endings can become shortcuts for a classifier even though they are not the target mechanisms. Retaining the quality field allows those artifacts to be measured rather than silently filtered.

The stable quality composition across splits means that the observed target shift is not explained by a large change in truncation or refusal prevalence. This distinction matters for causal interpretation: the test split is less hallucination-heavy, but not substantially cleaner in terms of the auxiliary quality labels.

Table 8. Quality-label audit.

Split	Good	Truncated	Incorrect refusal	Good percentage
Train	14,942	28	120	99.02%
Test	2,675	1	24	99.07%

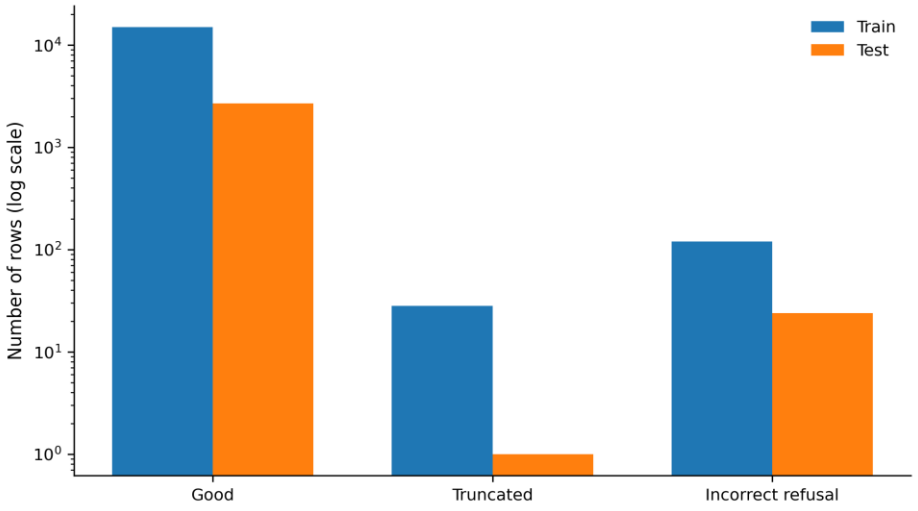

Figure 5. Quality-label distribution on a logarithmic scale.

4.4 Task and generator variation

RAGTruth spans several evidence structures rather than a single question-answering template. Table 9 reports the original release statistics by task, and Figure 6 compares total responses with hallucination responses. Data-to-

text contributes 6,198 responses, of which 4,254 contain hallucination annotations, as well as 9,290 hallucinated spans. Question answering contributes 5,934 responses and 1,724 hallucination responses. The two summarization sources have lower absolute burdens.

These differences caution against interpreting a single pooled score as task-invariant competence. Structured data-to-text generation can expose entity-value swaps, null-field inventions, and numeric contradictions that differ from narrative summarization errors. Question answering can depend on whether the retrieved passages jointly support the answer, while long source documents create context-selection pressure. Task-stratified metrics should therefore accompany the overall result, particularly when a detector is intended for a domain with one dominant evidence format.

Generator variation is similarly pronounced. Table 10 and Figure 7 summarize hallucination responses and spans by source model in the original release. GPT-4 and GPT-3.5 contribute fewer annotated hallucination responses than the Llama-2 and Mistral configurations, while Llama-2-13B has the highest span-to-response ratio in the table. A classifier can inadvertently exploit generator-specific phrasing, verbosity, or refusal patterns. Generator-stratified evaluation, leave-one-generator-out validation, and adversarial removal of model identifiers are therefore useful robustness checks.

Table 9. Original RAGTruth task-level release statistics.

Task	Instances	Responses	Hallucination responses	Hallucination spans
Summarization (CNN/DM)	628	3,768	1,165	1,474
Summarization (Recent News)	315	1,890	521	598
Question answering	989	5,934	1,724	2,927
Data-to-text	1,033	6,198	4,254	9,290
Overall	2,965	17,790	7,664	14,289

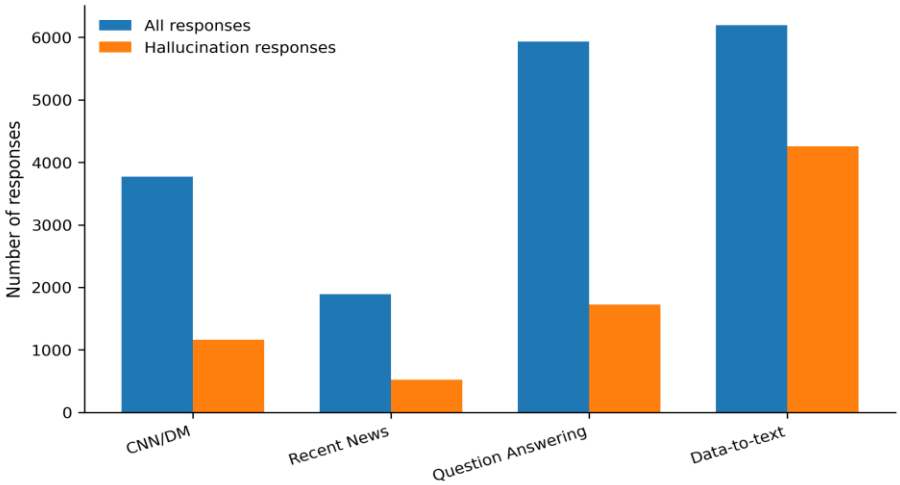

Figure 6. Original RAGTruth response and hallucination burden by task.

Table 10. Original generator-level hallucination statistics from RAGTruth.

Generator	Hallucination responses	Hallucination spans	Spans per hallucination response
GPT-3.5-turbo-0613	401	533	1.33
GPT-4-0613	406	485	1.19

Generator	Hallucination responses	Hallucination spans	Spans per hallucination response
Llama-2-7B-chat	1,832	3,302	1.80
Llama-2-13B-chat	1,677	3,799	2.27
Llama-2-70B-chat	1,395	2,608	1.87
Mistral-7B-Instruct	1,953	3,562	1.82

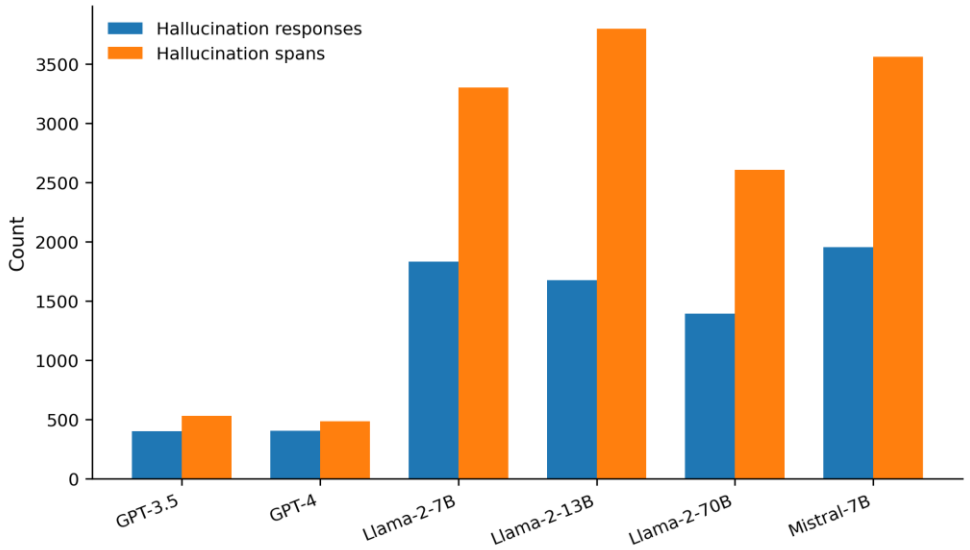

Figure 7. Original RAGTruth hallucination responses and annotated spans by generator.

4.5 Distribution tests

Table 11 reports the three split-composition tests. The four-class result is statistically significant (chi-square = 108.381, df = 3, $p = 2.447 \times 10^{-23}$) with Cramér's V = 0.078. The binary none-versus-any test is also significant (chi-square = 85.926, df = 1, $p = 1.868 \times 10^{-20}$) with V = 0.069. Both effect sizes are small, but the direction is consistent: the test split contains a larger share of none and a smaller share of mixed and baseless-only errors.

The quality-label test is different. Its chi-square value is 3.348 with df = 2 and p = 0.188; Cramér's V is 0.014. At the conventional 0.05 level, the split difference is not statistically significant. This supports the earlier interpretation that target composition shifts while auxiliary quality composition remains stable.

The small effect sizes do not eliminate the practical issue. A safety classifier is evaluated precisely on minority behavior, so a few percentage points of composition change can alter accuracy, precision, calibration, and threshold choice. Reported model comparisons should therefore use the same fixed split, preserve per-class metrics, and avoid choosing thresholds from the test distribution.

Table 11. Train-test distribution tests.

Audit target	Chi-square	df	p-value	Cramér's V	Interpretation
Four-class target	108.381	3	2.447×10^{-23}	0.078	Class composition differs with a small effect
Binary hallucination	85.926	1	1.868×10^{-20}	0.069	Overall prevalence differs with a small effect

Audit target	Chi-square	df	p-value	Cramér's V	Interpretation
Quality labels	3.348	2	0.188	0.014	No significant quality shift at 0.05

4.6 Operational implications for conflict-aware monitoring

The empirical structure of the benchmark leads to a mechanism-aware routing policy rather than a generic warning. Table 12 maps each predicted class to a primary diagnostic and corrective action. None is not treated as proof of truth; it indicates that the detector found no marked mechanism above the operating threshold. Evident conflict triggers evidence reconciliation. Baseless information triggers evidence expansion, claim deletion, or abstention. Both triggers the most conservative route.

Probability calibration determines whether this policy is usable. A system may choose a high-recall threshold for evident conflict in regulated reporting, a high-precision threshold for automatic answer rewriting, or a selective prediction rule that sends uncertain cases to review. Because confidence quality can vary by task and generator, calibration should be reported overall and by subgroup [27]. Evidence-card interfaces can then present the predicted mechanism, confidence, source passage, and implicated answer span in a compact human-review view [39].

The application literature reinforces this routing interpretation. Numeric RAG systems need explicit reconciliation of values and entities [31], [32]; operational assistants need provenance before emitting root-cause narratives [33], [34]; long-document contract analysis needs evidence across clauses rather than only local lexical overlap [35]; and ambiguous log or humanitarian cases may require selective refusal [36], [40]. Conflict-aware labels provide a common control vocabulary across these domains without requiring the downstream system to use the same generator or retriever.

Table 12. Mechanism-specific routing policy for a post-generation safeguard.

Predicted class	Primary risk	Recommended action	Escalation condition
none	Undetected error remains possible	Return answer with evidence display	Low confidence or high-risk domain
evident_conflict	Answer contradicts retrieved evidence	Locate conflict, reconcile source, regenerate constrained answer	Numeric, legal, medical, or policy claim
baseless_info	Answer exceeds available support	Expand retrieval, remove unsupported claim, or abstain	No supporting passage after second retrieval
both	Contradiction and unsupported addition coexist	Block automatic release and require full evidence review	Any high-confidence mixed prediction

5. Limitations

The study is centered on one benchmark. RAGTruth is diverse in task and generator coverage, but its annotation guidelines, source domains, prompt formats, and generator families do not represent every production RAG system. External validation is needed before a threshold or class prevalence is transferred to a different corpus. In particular, multilingual retrieval, multimodal evidence, conversational memory, and highly specialized numerical reasoning can introduce error patterns that are only partly represented here.

The four-class target is row-level. It preserves the distinction between contradiction and unsupported addition, but it compresses span count, span severity, and location into a single label. Two answers assigned to both may differ greatly: one may contain a small unsupported phrase plus a minor quantity conflict, while another may be unreliable throughout. Span-level auxiliary supervision, claim decomposition, and severity-aware scoring would provide finer explanations.

The model families also impose representational constraints. TF-IDF features can overvalue lexical novelty and undervalue faithful paraphrase. A BiLSTM trained from scratch may struggle with rare entities and long-range evidence alignment. A 512-token DeBERTa input can omit relevant context unless truncation or window aggregation is designed carefully. Long-context encoders such as those used by recent RAG hallucination detectors [15] provide a promising extension, but increase memory and latency relative to the compact baseline.

Finally, generator and task metadata can function either as useful covariates or as shortcuts. Including them may improve in-distribution prediction while reducing portability to unseen systems. The safest reporting practice is to separate a text-only primary model from metadata-augmented analysis and to include task-stratified, generator-stratified, and leave-one-generator-out results. Calibration should likewise be reassessed after any change to the retriever, evidence formatting, or upstream generator.

6. Conclusion

This study develops a conflict-aware formulation for RAG answer-quality classification that separates no marked hallucination, evident conflict, baseless information, and mixed errors. The distinction retains information that a binary detector discards and maps directly to different corrective actions. RAGTruth-processed supplies the necessary query, evidence, answer, metadata, and two-flag labels for this formulation.

The empirical audit establishes the conditions under which lightweight models should be judged. The train and test splits differ significantly but modestly in target composition; baseless-only errors are more common than evident-only errors; the both class is the smallest test category; and quality labels are stable across splits. Task and generator statistics reveal additional heterogeneity. These findings make macro-F1, class-specific recall, confusion matrices, AUPRC, calibration, and subgroup analysis more informative than accuracy alone.

The proposed XGBoost, BiLSTM, and DeBERTa-small designs span three practical operating points: inexpensive lexical screening, compact sequence modeling, and contextual semantic comparison. Their shared probability interface supports thresholding, selective prediction, and mechanism-specific routing. More broadly, conflict-aware monitoring turns hallucination detection from a generic alarm into an actionable evidence-control layer: contradictions can be reconciled, unsupported claims can be removed or retrieved, and mixed errors can be escalated before the answer reaches a user.

References

[1] C. Niu, Y. Wu, J. Zhu, S. Xu, K. Shum, R. Zhong, J. Song, and T. Zhang, “RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models,” in Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (ACL), Bangkok, Thailand, 2024, pp. 10862–10878, doi: 10.18653/v1/2024.acl-long.585.

[2] Weights & Biases, “RAGTruth-processed,” Hugging Face Datasets, 2024.

[3] ParticleMedia, “RAGTruth,” GitHub repository, 2024.

[4] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 9459–9474.

[5] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, “Retrieval Augmentation Reduces Hallucination in Conversation,” in Findings of EMNLP, 2021, pp. 3784–3803.

[6] Z. Ji et al., “Survey of Hallucination in Natural Language Generation,” ACM Comput. Surv., vol. 55, no. 12, pp. 1–38, 2023.

[7] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, “On Faithfulness and Factuality in Abstractive Summarization,” in Proc. ACL, 2020, pp. 1906–1919.

[8] W. Kryscinski, B. McCann, C. Xiong, and R. Socher, “Evaluating the Factual Consistency of Abstractive Text Summarization,” in Proc. EMNLP, 2020, pp. 9332–9346.

[9] P. Laban, T. Schnabel, P. N. Bennett, and M. A. Hearst, “SummaC: Re-Visiting NLI-Based Models for Inconsistency Detection in Summarization,” Trans. Assoc. Comput. Linguistics, vol. 10, pp. 163–177, 2022.

[10] Y. Zha, Y. Yang, R. Li, and Z. Hu, “AlignScore: Evaluating Factual Consistency with a Unified Alignment Function,” in Proc. ACL, 2023, pp. 11328–11348.

[11] P. Manakul, A. Liusie, and M. J. F. Gales, “SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models,” in Proc. EMNLP, 2023, pp. 9004–9017.

[12] S. Min et al., “FActScore: Fine-Grained Atomic Evaluation of Factual Precision in Long Form Text Generation,” in Proc. EMNLP, 2023, pp. 12076–12100.

[13] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, “RAGAS: Automated Evaluation of Retrieval Augmented Generation,” arXiv:2309.15217, 2023.

[14] J. Li, X. Cheng, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, “HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models,” in Proc. EMNLP, 2023, pp. 6449–6464.

[15] R. Zhang, Z. Wen, C. Wang, C. Tang, P. Xu, and Y. Jiang, "Quality analysis and evaluation prediction of RAG retrieval based on machine learning algorithms," arXiv preprint arXiv:2511.19481, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.19481

[16] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proc. KDD, 2016, pp. 785–794.

[17] 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, pp. 4171–4186.

[18] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-Enhanced BERT with Disentangled Attention,” in Proc. ICLR, 2021.

[19] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[20] M. Schuster and K. K. Paliwal, “Bidirectional Recurrent Neural Networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, 1997.

[21] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[22] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in Adv. Neural Inf. Process. Syst., vol. 32, 2019.

[23] T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” in Proc. EMNLP: System Demonstrations, 2020, pp. 38–45.

[24] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in Proc. ICLR, 2015.

[25] J. Davis and M. Goadrich, “The Relationship Between Precision-Recall and ROC Curves,” in Proc. ICML, 2006, pp. 233–240.

[26] T. Fawcett, “An Introduction to ROC Analysis,” Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006.

[27] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks,” in Proc. ICML, 2017, pp. 1321–1330.

[28] N. Thakur et al., “BEIR: A Heterogeneous Benchmark for Zero-Shot Evaluation of Information Retrieval Models,” in Proc. NeurIPS Datasets and Benchmarks, 2021.

[29] C. Li, W. Su, and E. Zhang, “Lightweight Hallucination Firewall for Enterprise LLM Applications: Evidence Consistency, Self-Checking, and Small-Model Detection on TruthfulQA,” JACS, vol. 3, no. 1, pp. 49–65, Jan. 2023, doi: 10.69987/JACS.2023.30104.

[30] C. Li, J. Bai, and S. Wang, “Evidence-Chain Reliable RAG: Word-Level Hallucination Detection, Source Attribution, and Provenance Explanation for LLM Applications,” JACS, vol. 4, no. 2, pp. 76–92, Feb. 2024, doi: 10.69987/JACS.2024.40207.

[31] K. Zhang, S. Meng, and E. Zhou, “Evidence-Grounded Trading Desk Risk Memos over SEC Filings: Retrieval-Augmented Generation with XBRL Numeric Verification,” JACS, vol. 3, no. 2, pp. 60–76, Feb. 2023, doi: 10.69987/JACS.2023.30205.

[32] Q. Wu, J. Bai, and X. Zhou, “Evidence-Grounded Financial RAG: Reducing Numerical Hallucination in LLM-Generated Corporate Risk Memos,” JACS, vol. 3, no. 3, pp. 65–84, Mar. 2023, doi: 10.69987/JACS.2023.30306.

[33] B. Zhang, H. Rao, and D. Zhao, "Evidence-Grounded RAG for Cloud-Native DevOps: Hallucination-Resistant AIOps Question Answering over Private Operations Documents," JACS, vol. 4, no. 3, pp. 109–125, Mar. 2024, doi: 10.69987/JACS.2024.40308.

[34] G. Liu, C. Li, and E. Zhang, "OpsLLM for Cloud Incident Triage: Bilingual RAG-Based Root Cause Analysis and Alert Summarization for AI Infrastructure Operations," JACS, vol. 4, no. 4, pp. 97–111, Apr. 2024, doi: 10.69987/JACS.2024.40408.

[35] S. Zhou, Z. Li, and E. Wang, "Long-Document RAG for Contractual and Insurance Clause Analysis in Receivables RWA Structures," JACS, vol. 4, no. 8, pp. 88–104, Aug. 2024, doi: 10.69987/JACS.2024.40810.

[36] J. Nie and D. Zheng, "Ambiguity-Aware HDFS Log Anomaly Detection with Retrieval-Augmented Failure Narratives and Selective Refusal," JACS, vol. 3, no. 1, pp. 66–80, Jan. 2023, doi: 10.69987/JACS.2023.30105.

[37] D. Zheng, B. Zhang, and J. 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.

[38] W. Su, S. Chen, and C. Zhao, "Budgeted Multi-Hop Retrieval Agent for Compositional Question Answering: A Retrieval-Policy Evaluation on the Official MultiHop-RAG Benchmark," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 649–662, Dec. 2025, doi: 10.51903/jtie.v4i3.543.

[39] J. Jin, "LLM-Style Evidence Cards for Scientific Search Interfaces: A UI/UX Design Framework for Retrieval Transparency, Ranking Trust, and Visual Evidence Hierarchy," Int. J. Graph. Des., vol. 3, no. 2, pp. 397–414, Oct. 2025, doi: 10.51903/ijgd.v3i2.3698.

[40] Y. Chen and H. Xu, "Trust-Calibrated Multilingual RAG for Humanitarian Information Platforms: Empirical Evaluation on OMoS-QA for Migration Information Access," Int. J. Graph. Des., vol. 4, no. 1, pp. 141–164, Apr. 2026, doi: 10.51903/ijgd.v4i1.3552.