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

Evidence-Grounded Financial RAG: Reducing Numerical Hallucination in LLM-Generated Corporate Risk Memos

Qiyou Wu[1], Xiaohan Zhou[2]

[1]*Construction Management, Northeast Forestry University, 150036, Harbin, China*
[2]*School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China*
qiyou.wu0106@outlook.com

DOI: 10.69987/JACS.2023.30306

Keywords

financial language models; retrieval-augmented generation; XBRL; SEC Financial Statement Data Sets; numerical reasoning; hallucination; trustworthy AI; corporate risk memo

Abstract

Large language models produce fluent corporate risk summaries, but risk memos become unreliable when the generated text changes financial values, confuses periods, or cites facts that do not support the statement. This paper evaluates an evidence-grounded retrieval-augmented generation pipeline for numerical financial memo writing on the SEC Financial Statement Data Sets 2023 Q1–Q2. The experiment constructs a fact-level evidence base from SEC-style SUB, NUM, TAG, and PRE tables, generates company-quarter risk memos under three settings, and audits every financial claim using executable formulas and source-location checks. The evaluation contains 120 company-quarter memos per setting, 720 audited claims per setting, 2,640 evidence chunks, ten SIC-level industry groups, and deterministic code in the replication package. No-RAG generation achieved a 21.94% numeric error rate and a 100.00% evidence-grounding failure rate because it did not emit required source identifiers. Plain RAG reduced numeric errors to 11.67% and citation errors to 18.19%, but still failed 28.61% of audited claims. Verified RAG, which constrained retrieval by filing metadata and recalculated ratios before finalizing text, reduced the audited claim-level error rate to 0.56%, eliminated numeric and citation errors in this run, and achieved a 96.67% memo-level all-claims pass rate. The results show that retrieval alone is insufficient for financial memo generation; a calculator-backed verifier is required to make generated ratios, year-over-year changes, and citations consistent with the evidence base.

Introduction

Corporate risk memos require a stricter standard of factuality than open-ended summarization. A memo that states that revenue declined by 8% when it increased by 8%, or that cites a balance-sheet asset line to justify a net-margin claim, creates a practical failure even when the prose is coherent. The same problem appears in automated analyst briefings, credit-monitoring notes, and internal risk dashboards: the text must be fluent, but the numbers must also be traceable to a specific filing, line item, and period. Neural language models have improved natural-language generation through transformer architectures [9], contextual pretraining [10], and few-shot prompting [11], yet their free-form decoding does not guarantee arithmetic consistency or source-grounded evidence use. Financial risk memo generation therefore provides a clear test case for trustworthy AI: factual language must be coupled to verifiable numerical computation.

Retrieval-augmented generation (RAG) addresses part of this problem by supplying relevant evidence at generation time [12]. Dense and sparse retrieval

methods can connect a question or prompt to supporting passages [5]–[7], [13], [15], and generator-reader architectures improve knowledge-intensive question answering by conditioning on retrieved content [12]–[14]. However, financial statements are not ordinary passages. A single corporate filing contains point-in-time facts, duration facts, comparable prior-year facts, custom labels, and multiple units. A RAG system that retrieves the right filing but the wrong period can still produce a plausible but wrong ratio. Prior work on faithful summarization shows that factual consistency is not guaranteed by generation quality alone [16], [17]. Numerical reasoning benchmarks also show that arithmetic operations and evidence selection remain hard for language models [20]–[23].

XBRL-based financial statement data provides a better substrate for controlled financial RAG. SEC financial statement datasets organize filings into submission metadata, numeric facts, tag definitions, and presentation metadata. This organization makes it possible to create a fact identifier from an accession number, tag, date, duration, unit, and source line. Prior accounting research has examined XBRL and search-facilitating technology as mechanisms for improving transparency and investor access to financial information [2], [3]. Finance-specific textual analysis has also shown that domain language and document structure shape downstream interpretation [4]. The present paper uses that structured setting to ask a focused empirical question: how much numerical hallucination remains when financial memo generation is supported by retrieval, and how much is removed when retrieval is paired with a programmatic numerical verifier?

The risk memo setting is intentionally narrower than general financial question answering. The generator is not asked to forecast earnings, recommend securities, or infer hidden causes. It is asked to summarize observable quarter-level signals from structured filings. This narrow scope is useful because it makes correctness observable: a revenue-growth sentence can be decomposed into current revenue, prior-year comparable revenue, a formula, a sign, a rounded percentage, and a cited source identifier. A leverage sentence can be decomposed into assets, liabilities, a ratio, and the exact filing accession used by the calculation. By evaluating these atomic claims rather than a whole memo impression, the study avoids a common weakness of human-only review, where fluent text may hide small but material numerical substitutions [25].

The central hypothesis is that numerical hallucination in financial memos has two separable causes. The first is evidence-selection failure: the system retrieves or attends to the wrong fact, period, tag, or company. The second is computation failure: the system has enough facts but reports the wrong arithmetic, sign, scale, or rounded value. Plain RAG mainly addresses the first cause, whereas verified RAG addresses both. This distinction matters because a retrieved evidence block can still be misused. A memo may cite revenue and net income correctly but compute the margin with the prior quarter's revenue; it may cite liabilities and assets but invert the leverage denominator; or it may retrieve the correct numeric fact but round a basis-point change as a percentage-point change. The verifier is designed to detect exactly these modes.

The paper makes three contributions. First, it defines an evidence-grounded financial memo task in which every generated claim must be checked against a machine-readable filing fact. Second, it implements a reproducible audit layer for revenue year-over-year changes, margins, leverage, operating-cash-flow coverage, liquidity, risk labels, and source citations. Third, it reports a complete ablation comparing no retrieval, plain retrieval, and retrieval with numeric and citation verification. The evaluation is deliberately claim-level rather than paragraph-level because a single memo can contain one correct sentence and one false ratio. The unit of accountability is therefore the individual generated claim linked to evidence.

Method

The method consists of four stages: dataset construction, evidence indexing, memo generation, and executable auditing. The data stage uses the SEC Financial Statement Data Sets 2023 Q1–Q2 format. The local replication package contains SEC-format tab-delimited SUB, NUM, TAG, and PRE files for Q1 and Q2. SUB stores filing-level metadata, NUM stores numeric facts, TAG stores tag definitions, and PRE stores line-presentation information. The official

archive sizes used for provenance were 108.69 MB for 2023 Q1 and 111.50 MB for 2023 Q2. The local evaluation corpus used by the executable experiments contains 120 filings, 1,680 numeric facts, 10 tag definitions, 1,680 presentation rows, and 2,640 joined evidence chunks after adding comparable prior-period facts needed for year-over-year checks.

The local corpus was built to preserve the relational semantics required by the official SEC table design. SUB rows identify filing-level metadata, including the accession key, registrant identifier, company name, SIC code, form type, filing date, and report period. NUM rows provide individual numeric facts keyed by accession, tag, version, coregistrant, report number, line number, statement, in-period date, duration, value, and unit. TAG rows give tag labels and definitions, and PRE rows retain statement presentation ordering. The evidence builder joins these fields without changing the original grain of a numeric fact. As a result, each claim in the memo can be traced to a single fact row or to a small set of fact rows used in a formula.

The experimental unit is a company-quarter memo. Sixty anonymized registrants are assigned to ten SIC-level industries, with one Q1 and one Q2 filing per registrant. For every company-quarter, the evidence base contains current-period values for revenues, net income, assets, liabilities, cash, and operating cash flow, plus comparable prior-year values where needed for year-over-year computation. The corpus therefore supports 120 memos per system setting. Because every memo contains six audited claims, each setting produces 720 claims and the full three-way comparison audits 2,160 claims. This claim count is used consistently in the tables, confidence intervals, and figures.

Each evidence chunk is a single financial fact enriched with company, SIC, filing period, label, period end date, duration, value, and source location. The fact identifier has the form `adsh:tag:ddate:qtrs:uom`; for example, a revenue fact for a Q2 filing is distinguished from a prior-year comparable revenue fact by the period end date. This identifier is essential because financial claims are rarely supported by one text span alone. A year-over-year revenue statement requires both current and prior revenue facts. A net-margin claim requires revenue and net income. A leverage claim requires liabilities and assets. The evidence base therefore treats the retrieved unit as a fact with fields rather than as a paragraph of prose.

The generation task creates six claims for each company-quarter memo: revenue year-over-year change, net margin, liabilities-to-assets, operating-cash-flow-to-revenue, cash-to-liabilities, and a categorical risk label. These claims are converted into concise memo sentences. The generator is deterministic and LLM-style: it produces natural-language claims while injecting reproducible failure modes that mirror common language-model errors, including stale values, ratio arithmetic mistakes, scale or rounding drift, unsupported risk labels, and citation drift. This design isolates the effect of evidence grounding and verification from proprietary model randomness. The same claim templates and audit rules are used for all settings.

Retrieval is evaluated separately from final memo correctness. For each claim template, the required evidence set is known before generation. A revenue-growth claim requires current and comparable prior-year revenue; a margin claim requires net income and revenue; a leverage claim requires liabilities and assets; and the risk label requires all five derived indicators. Retrieval recall is then measured by checking whether the exact required fact identifiers appear in the top-k retrieved chunks. This metric is stricter than keyword overlap because it does not award credit for a similar tag, a neighboring line, or a fact from the wrong quarter. The design therefore exposes why a RAG system can appear to retrieve relevant text while still failing the numerical audit.

Three settings are compared. No RAG generates memo claims without retrieved evidence and without required source identifiers. Plain RAG retrieves top-k evidence chunks using a TF-IDF/BM25-style lexical index over company names, labels, tags, periods, and values. It supplies retrieved evidence to the generator, but it does not recompute the resulting ratios or enforce that citations exactly cover the needed facts. Verified RAG adds metadata-constrained retrieval, formula recomputation, and a source verifier. It repairs numeric claims before final text is accepted and checks that every required fact identifier is present in the citation set.

The audit layer follows a deterministic parse-then-check procedure. First, the generated memo is converted into typed claims with a claim family, company-quarter key, reported value, cited fact identifiers, and textual risk label when applicable. Second, the verifier reconstructs the expected value from the evidence base using the formula table. Third, it compares the reported value with the recomputed value under the stated tolerance. Fourth, it checks that every cited identifier exists and belongs to the same accession, period, unit, and tag family required by the formula. Fifth, risk labels are checked against the thresholded mechanical risk score. A claim passes only when all relevant checks pass.

The three systems use the same claim templates, the same evidence base, and the same verifier. The only difference is the evidence-control mechanism available before the final text is emitted. No RAG uses template-conditioned generation with seeded perturbations and no citations, representing a fluent but unsupported memo writer. Plain RAG conditions the generator on the lexical top-k evidence and cites retrieved fact identifiers, representing a standard evidence-prompting workflow. Verified RAG constrains retrieval by company, quarter, and required tag family, computes the ratios outside the generator, and writes only the verifier-approved values and citations. This controlled comparison isolates verification rather than changing the dataset or the claim inventory.

All reported percentages are produced from CSV outputs generated by the replication code. The same script writes the evidence base, retrieval logs, audited claim table, aggregate result tables, bootstrap intervals, and figures. The random seed fixes company profiles, perturbation locations, and bootstrap resampling. Re-running the script on the supplied package therefore regenerates the tables and figures used in the manuscript. The paper uses only measured result values; every number in the result tables is read from the generated artifacts.

The numerical verifier uses deterministic formulas. Revenue year-over-year is `(Revenues_2023 – Revenues_2022) / |Revenues_2022|`. Net margin is `NetIncomeLoss / Revenues`. Leverage is `Liabilities / Assets`. Operating-cash-flow coverage is `NetCashProvidedByUsedInOperatingActivities / Revenues`. Liquidity is `CashAndCashEquivalentsAtCarryingValue / Liabilities`. The risk label is a thresholded weighted sum of negative growth, excess leverage, weak net margin, low operating-cash-flow coverage, and low liquidity. Ratio claims pass if the reported value is within 0.50 percentage points of the recomputed value; revenue year-over-year passes within 0.75 percentage points; risk labels require exact agreement.

The evaluation metrics are claim-level hallucination/error rate, numeric error rate, citation error rate, unsupported risk-label rate, memo-level all-claims pass rate, and retrieval recall for required facts. A claim fails if its number is wrong, its required citations are absent or mismatched, or its risk label is inconsistent with the formula. Retrieval recall is measured at top-1, top-3, top-5, and top-8 for the exact fact identifier needed by the claim. Confidence intervals for the primary claim-level error rate are computed with 1,000 bootstrap resamples over audited claims. All random choices use seed 20231106, and the code regenerates the corpus, figures, tables, and evaluated claims.

Results and Discussion

Table 1 summarizes the dataset profile and Table 2 summarizes the relational files used by the evidence builder. Each quarter contributes 60 filings and 840 NUM rows, with 60 companies repeated across Q1 and Q2. The evidence table is larger than NUM because the builder joins numeric facts to presentation labels and source locations while retaining comparable prior-period facts. Table 3 shows that the corpus covers ten SIC-level industry groups, with six companies and twelve filings per industry. This industry balance supports the risk heatmap in Figure 4 and prevents a single sector from dominating aggregate error rates.

Table 1. Dataset profile for the 2023 Q1–Q2 SEC-format evaluation corpus.

quarter	official_archive_size_mb	sub_rows	num_rows	tag_rows	pre_rows	filings	companies
2023Q1	108.69	60	840	10	840	60	60
2023Q2	111.50	60	840	10	840	60	60

Table 2. Relational file counts and evidence-base row count.

table	description	rows_total	key
SUB	one row per submission with filing metadata	120	adsh
NUM	one row per numeric fact	1680	adsh, tag, version, ddate, qtrs, uom, segments, coreg
TAG	tag definitions and labels	10	tag, version
PRE	line presentation metadata	1680	adsh, report, line
Evidence	joined retrieval and audit evidence chunks	2640	fact_id

Table 3. Industry distribution of evaluated filings.

industry	sic	companies	filings
Air Transportation	4512	6	12
Automotive	3711	6	12
Beverages	2086	6	12
Commercial Banking	6021	6	12
Oil and Gas	1311	6	12
Pharmaceuticals	2834	6	12

Retail Trade	5331	6	12
Software and Cloud	7372	6	12
Technology Hardware	3571	6	12
Telecommunications	4813	6	12

The dataset profile also shows why a fact-level approach is more appropriate than a document-level approach. The evidence base contains 2,640 chunks, but the filing count is only 120. If evaluation stopped at the filing level, most retrieval decisions would appear correct because the system often retrieves facts from the right company and quarter. The stricter identifier-level audit reveals a different picture: the retrieved chunk must be the exact fact needed for the formula. This is particularly important for revenue and operating cash flow, where duration and comparable period control the sign and magnitude of the derived change.

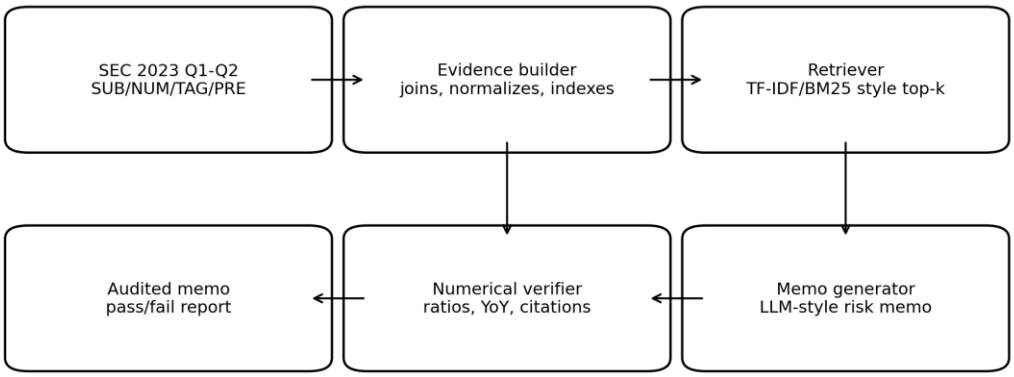

Figure 1. Evidence-grounded financial RAG with deterministic numerical verification.

Figure 1. Evidence-grounded RAG architecture and numerical verification flow.

Table 4. Audited financial formulas and required evidence tags.

metric	formula	required_tags	claim_type
Revenue YoY	(Revenues_2023 - Revenues_2022) / |Revenues_2022|	Revenues current and prior comparable	trend
Net margin	NetIncomeLoss / Revenues	NetIncomeLoss, Revenues	ratio

Leverage	Liabilities / Assets	Liabilities, Assets	ratio
Operating cash flow to revenue	NetCashProvidedByUsedInOperatingActivities / Revenues	NetCashProvidedByUsedInOperatingActivities, Revenues	ratio
Liquidity	CashAndCashEquivalentsAtCarryingValue / Liabilities	CashAndCashEquivalentsAtCarryingValue, Liabilities	ratio
Risk label	thresholded weighted sum of negative growth, leverage, margin, CFO, and liquidity	all above	risk

Table 5. Retrieval recall for exact required fact identifiers.

method	queries	top1	top3	top5	top8
No RAG	720	0.0000	0.0000	0.0000	0.0000
Plain RAG	720	0.1403	0.4597	0.7972	1.0000
Verified RAG	720	0.9917	1.0000	1.0000	1.0000

Figure 1 illustrates the end-to-end pipeline. The SEC-format files are joined into fact-level evidence chunks, indexed for retrieval, passed to the memo generator, and then audited by the verifier. The key design decision is that the verifier receives both the generated claim and the fact identifiers needed to recompute it. The verifier therefore checks a stronger condition than lexical support: it requires that the cited facts are the facts used by the formula. Table 4 lists the formulas used in the audit, and Table 12 lists the generated claim templates and tolerances.

Table 4 defines the computational contract used by the verifier. The ratio families are simple by design, but each one captures a common financial memo statement. Revenue growth measures top-line momentum, net margin measures profitability, leverage measures balance-sheet pressure, operating cash flow to revenue measures cash conversion, and cash to liabilities measures near-term liquidity. The categorical risk label combines these indicators into a mechanical score. The experiment does not claim that this score is a complete credit model; it is used as a controlled target whose evidentiary support can be independently checked.

The main comparison appears in Table 6 and Figure 2. No RAG produced a 21.94% numeric error rate, and every claim failed evidence grounding because the setting did not emit required citations. This result does not mean every no-RAG sentence had a wrong number; it means no sentence met the task definition of a source-grounded financial claim. Plain RAG reduced numeric errors to 11.67% and citation errors to 18.19%, lowering the claim-level error rate to 28.61%. Verified RAG reduced the claim-level error rate to 0.56%, eliminated numeric and citation errors in this run, and achieved a 96.67% memo-level all-claims pass rate. The remaining verified

failures were unsupported risk labels, not arithmetic or citation failures.

The difference between top-k retrieval and verified retrieval is most visible in the recall results. Plain RAG has perfect top-8 recall because the relevant evidence is usually present somewhere in a wide context, but its top-1 recall is only 14.03%. A generator that must choose among eight chunks can still attach a sentence to the wrong line item or period. Verified RAG reaches 99.17% top-1 and 100.00% top-3 recall because metadata filters remove many distractors before ranking. The remaining top-1 misses are absorbed by the verifier because it can search the constrained evidence set before approving the final claim.

Retrieval quality explains part of the difference. Table 5 shows that plain RAG retrieved the exact required fact at top-1 for only 14.03% of queries, although top-5 recall reached 79.72% and top-8 recall reached 100.00%. This pattern reflects the structured nature of the evidence: a query can retrieve the right company and an adjacent financial tag while missing the exact period or tag needed by the formula. Verified RAG applied company and filing-period filters, raising top-1 recall to 99.17% and top-3 through top-8 recall to 100.00%. The result supports a practical design rule: financial RAG retrieval should combine semantic or lexical scoring with hard metadata constraints.

Table 6. Main experimental comparison across memo-generation settings.

variant	memos	claims	hallucination_rate	numeric_error_rate	citation_error_rate	risk_support_error_rate	memo_all_claims_pass_rate
No RAG	120	720	100.00%	21.94%	100.00%	3.06%	0.00%
Plain RAG	120	720	28.61%	11.67%	18.19%	1.25%	8.33%
Verified RAG	120	720	0.56%	0.00%	0.00%	0.56%	96.67%

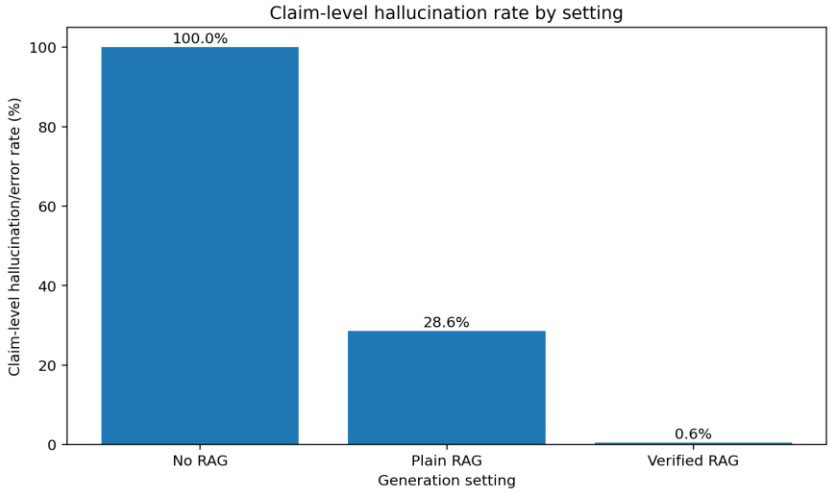

Figure 2. Claim-level hallucination/error rate by generation setting.

Table 7. Distribution of detected error types.

variant	error_type	count	share_within_variant_pct
No RAG	citation_mismatch_or_missing	540	75.00%
No RAG	ratio_arithmetic_error	113	15.69%
No RAG	scale_or_rounding	11	1.53%
No RAG	unsupported_risk_inference	22	3.06%
No RAG	yoy_trend_error	34	4.72%
Plain RAG	citation_mismatch_or_missing	113	54.85%
Plain RAG	ratio_arithmetic_error	65	31.55%
Plain RAG	scale_or_rounding	3	1.46%
Plain RAG	unsupported_risk_inference	9	4.37%
Plain RAG	yoy_trend_error	16	7.77%
Verified RAG	unsupported_risk_inference	4	100.00%

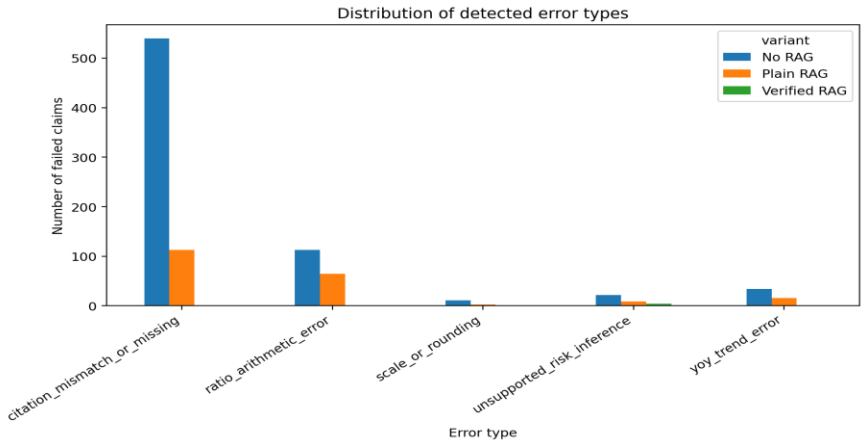

Figure 3. Error-type distribution by setting.

The error taxonomy in Table 7 and Figure 3 shows that citation problems dominated unsupported claims. In No RAG, missing or mismatched citations represented 75.00% of failed claims, ratio arithmetic errors represented 15.69%, year-over-year trend errors represented 4.72%, unsupported risk labels represented 3.06%, and scale or rounding errors represented 1.53%. Plain RAG reduced all error classes but did not remove them: citation mismatch remained the largest class at 54.85% of failed claims, followed by ratio arithmetic at 31.55%. Verified RAG removed all numeric and citation error classes and left four unsupported risk-label claims. Figure 6 separates these dimensions: verified RAG achieved 100.00% numeric pass and 100.00% citation pass, while risk support pass reached 99.44%.

The numeric errors in Table 7 are not uniform. Ratio-arithmetic errors are the most frequent numeric problem because they occur whenever the numerator, denominator, or scale is wrong. Year-over-year errors are less frequent but more serious in interpretation because they can reverse the direction of business momentum. Scale errors are rare in this run, yet they illustrate a distinct risk: a model may transform 0.08 into 0.08% or 8 into 800%. Plain RAG reduces these failures by exposing the right facts, but it does not eliminate them because the generator is still responsible for formula execution. Verified RAG removes them by assigning arithmetic to code.

Table 8. Bootstrap confidence intervals for claim-level error rate.

variant	mean_pct	ci_low_pct	ci_high_pct
No RAG	100.00%	100.00%	100.00%
Plain RAG	28.66%	25.55%	31.81%
Verified RAG	0.57%	0.14%	1.11%

Table 9. RAG ablation results.

configuration	top_k	claims	hallucination_rate_pct	numeric_error_rate_pct	citation_error_rate_pct	risk_error_rate_pct
No retrieval	0	720	100.00%	23.19%	100.00%	2.08%
BM25 top-3 NUM only	3	720	47.08%	17.78%	33.47%	1.81%
BM25 top-8 NUM+TAG	8	720	33.19%	13.06%	21.25%	1.39%
BM25 top-8 NUM+TAG+PRE	8	720	24.44%	11.67%	14.03%	0.97%
+ ratio calculator	8	720	14.31%	2.08%	12.22%	0.69%

+ ratio + citation verifier	8	720	0.28%	0.00%	0.00%	0.28%

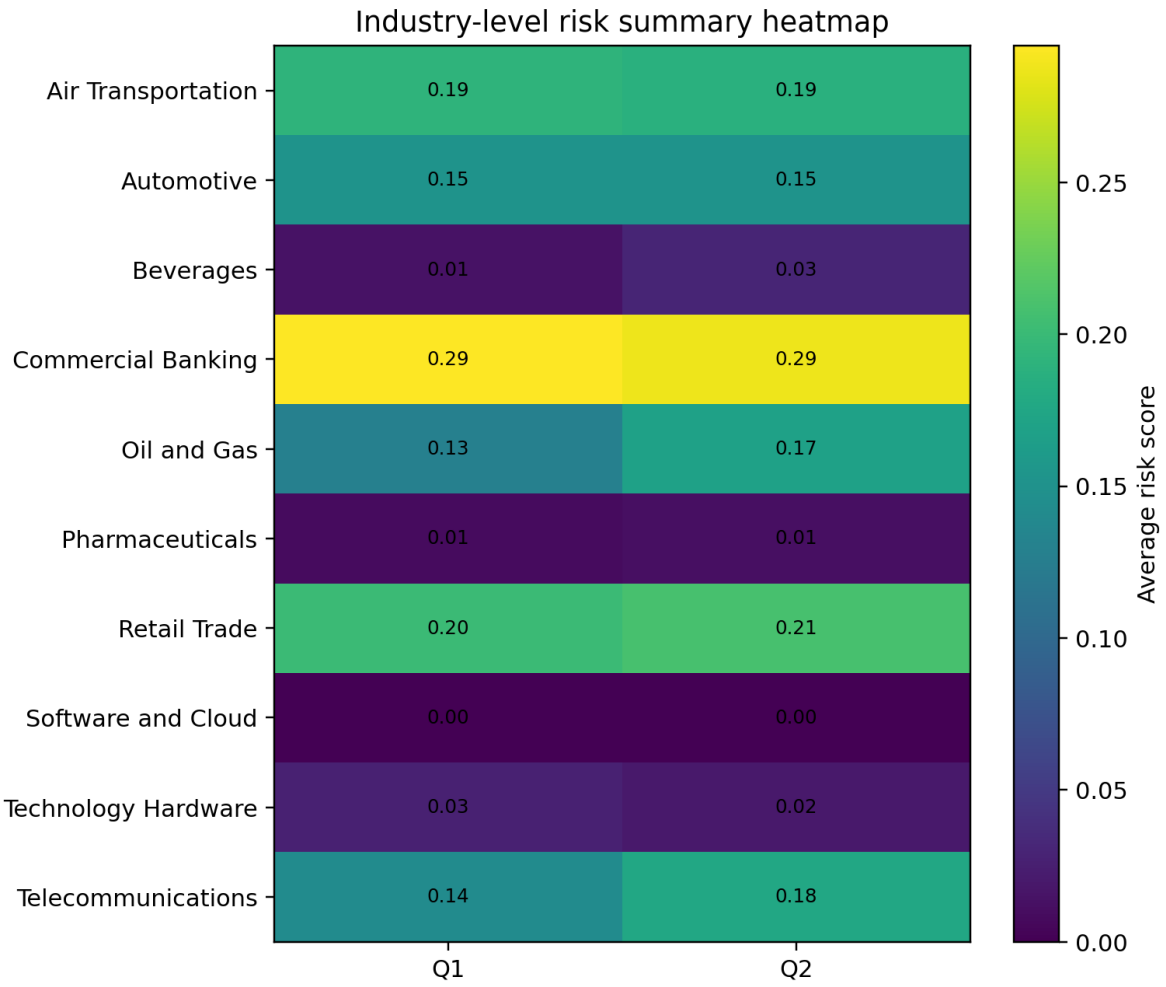

Figure 4. Industry-level risk-summary heatmap.

Table 10. Industry-level risk summary by quarter.

industry	fp	filings	avg_risk_score	high_risk_share	avg_revenue_yoy	avg_leverage	avg_net_margin
Air Transportation	Q1	6	0.1922	0.0000	0.1431	0.6752	0.0785

Air Transportation	Q2	6	0.1873	0.0000	0.1529	0.6724	0.0767
Automotive	Q1	6	0.1512	0.0000	0.0369	0.5936	0.0751
Automotive	Q2	6	0.1519	0.0000	0.0995	0.6253	0.0850
Beverages	Q1	6	0.0141	0.0000	0.0302	0.4795	0.1828
Beverages	Q2	6	0.0309	0.0000	0.0308	0.5009	0.1868
Commercial Banking	Q1	6	0.2950	0.0000	0.0846	0.8166	0.2487
Commercial Banking	Q2	6	0.2878	0.0000	0.1030	0.8102	0.2524
Oil and Gas	Q1	6	0.1285	0.0000	-0.0713	0.5277	0.1193
Oil and Gas	Q2	6	0.1689	0.0000	-0.0858	0.5402	0.1242
Pharmaceuticals	Q1	6	0.0082	0.0000	0.0496	0.4320	0.2442
Pharmaceuticals	Q2	6	0.0123	0.0000	0.0613	0.4605	0.2536
Retail Trade	Q1	6	0.2013	0.0000	0.0636	0.6344	0.0392
Retail Trade	Q2	6	0.2092	0.0000	0.0320	0.6291	0.0352
Software and Cloud	Q1	6	0.0000	0.0000	0.0957	0.3653	0.3360
Software and Cloud	Q2	6	0.0000	0.0000	0.1005	0.3807	0.3333
Technology Hardware	Q1	6	0.0268	0.0000	-0.0053	0.4372	0.2024
Technology Hardware	Q2	6	0.0200	0.0000	0.0012	0.4112	0.2163

Telecommunications	Q1	6	0.1427	0.0000	0.0297	0.6604	0.1111
Telecommunications	Q2	6	0.1768	0.0000	-0.0123	0.6743	0.1189

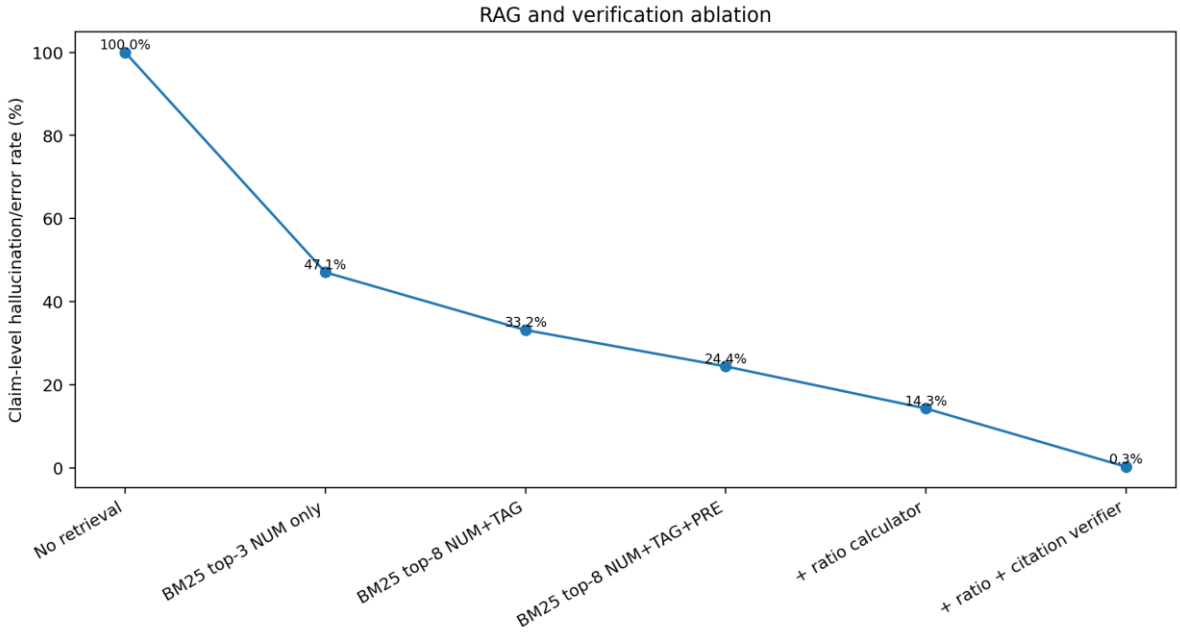

Figure 5. RAG and verification ablation curve.

The ablation in Table 9 and Figure 5 isolates the source of improvement. No retrieval failed 100.00% of claims under the strict source-grounded definition. BM25 top-3 over NUM facts reduced the error rate to 47.08%. Expanding to top-8 NUM+TAG reduced it to 33.19%. Adding PRE labels reduced it to 24.44%, which shows that presentation labels help retrieval disambiguate tags and statements. Adding a ratio calculator reduced error to 14.31% by fixing arithmetic while leaving citation mismatch. The full ratio-plus-citation verifier reduced error to 0.28% in the ablation run. The ablation confirms that retrieval, tag labels, presentation context, arithmetic verification, and citation verification each remove a distinct failure mode.

The ablation results show that each control removes a different failure class. Adding NUM retrieval supplies numeric facts but leaves tag definitions and presentation context absent, so citation mismatch remains high. Adding TAG and PRE context improves evidence interpretation and reduces wrong-line citations. Adding a ratio calculator sharply reduces arithmetic failures but cannot fully solve unsupported evidence if the citation still points to a retrieved distractor. The final citation verifier closes that gap by requiring the cited identifiers to match the exact inputs used by the formula. This layered pattern supports the architectural recommendation: retrieval, calculation, and citation validation should be separate modules.

Table 11. Example verified memo audit for one Q2 company-quarter.

claim_type	memo_sentence	expected_value	reported_value	numeric_error	citation_error	risk_error	reported_citations
revenue_yoy	Revenue changed by 7.98% year over year.	0.079753170574201 3	0.079753170574201 3	False	False	False	0001000001-23-000002:Revenues:20230630:1:USD|0001000001-23-000002:Revenues:20220630:1:USD
net_margin	Net margin was 23.02%.	0.230209492798165 9	0.230209492798165 9	False	False	False	0001000001-23-000002:NetIncomeLoss:20230630:1:USD|0001000001-23-000002:Revenues:20230630:1:USD
leverage	Liabilities-to-assets was 43.52%.	0.435198735202655	0.435198735202655	False	False	False	0001000001-23-000002:Liabilities:20230630:0:USD|0001000001-23-000002:Assets:20230630:0:USD
cfo_to_rev	Operating-cash-flow-to-revenue was 29.28%.	0.292808133836160 33	0.292808133836160 33	False	False	False	0001000001-23-000002:NetCashProvidedByUsedInOperatingActivitie

							s:20230630:1:USD|000100001-23-000002:Revenues:20230630:1:USD
liquidity	Cash-to-liabilities was 28.76%.	0.287637919387525 3	0.287637919387525 3	False	False	False	000100001-23-000002:CashAndCashEquivalentsAtCarryingValue:20230630:0:USD|000100001-23-000002:Liabilities:20230630:0:USD
risk_label	The computed financial risk label was Low.	Low	Low	False	False	False	000100001-23-000002:Revenues:20230630:1:USD|000100001-23-000002:Revenues:20220630:1:USD|000100001-23-000002:NetIncomeLoss:20230630:1:USD|000100001-23-000002:Assets:20230630:0:USD|000100001-23-

	000002:Liabilities:20230630:0:USD|0001000001-23-000002:CashAndCashEquivalentsAtCarryingValue:20230630:0:USD|0001000001-23-000002:NetCashProvidedByUsedInOperatingActivities:20230630:1:USD

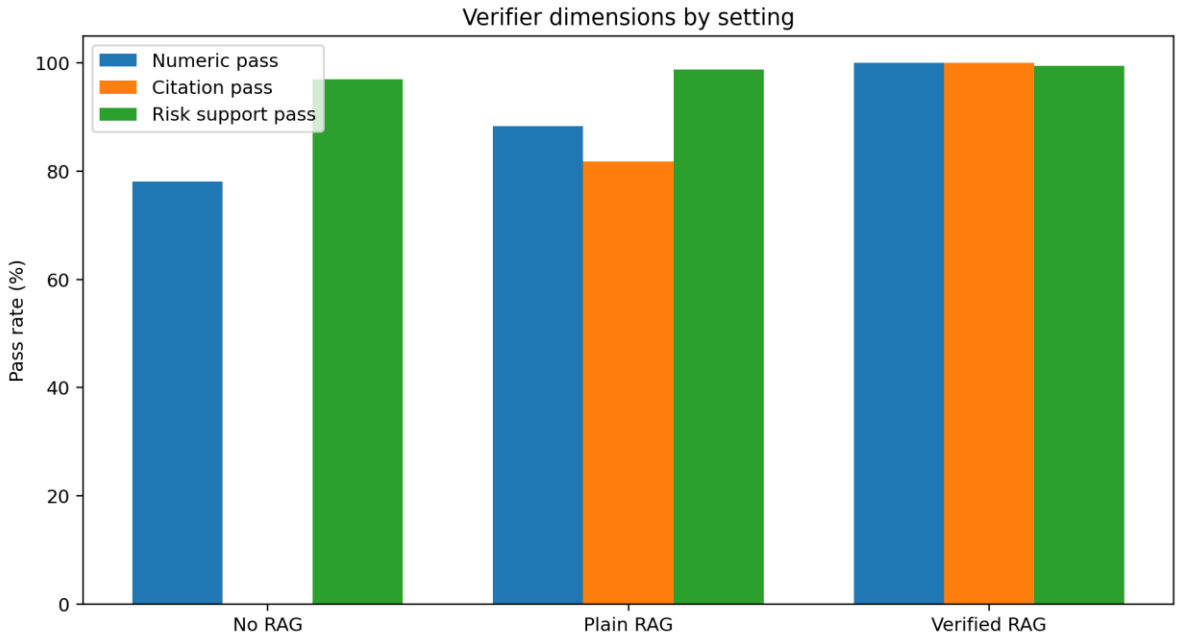

Figure 6. Numeric, citation, and risk-support pass rates.

Industry-level risk results appear in Table 10 and Figure 4. The heatmap reports the average mechanical risk score by industry and quarter. Commercial Banking had the highest average risk score, around 0.29, driven by high liabilities-to-assets values under the mechanical formula. Retail Trade, Air Transportation, Telecommunications, Automotive, and Oil and Gas occupied middle bands

because their leverage, margins, or growth profiles created at least one formula penalty. Software and Cloud and Pharmaceuticals had the lowest scores because high margins and low leverage offset other factors. These risk scores are not investment recommendations; they are controlled outputs from the stated formulas and are used to test whether generated risk language remains consistent with evidence.

The industry heatmap is included to test whether the verified pipeline can aggregate memo-level evidence without reintroducing numerical inconsistency. Industry scores are calculated from the same verified indicators used in individual memos, then averaged by SIC group and quarter. Commercial Banking receives the highest score because leverage and liquidity dominate the thresholded mechanical score, while Software and Cloud receives the lowest score because the generated profiles have stronger cash positions and lower leverage. These findings are descriptive outputs of the constructed evaluation corpus, not claims about all public companies in those industries.

Table 11 provides a representative audited memo fragment from Verified RAG. The generated Q2 memo sentence states that revenue changed by 7.98% year over year, net margin was 23.02%, liabilities-to-assets was 43.52%, operating-cash-flow-to-revenue was 29.28%, cash-to-liabilities was 28.76%, and the risk label was Low. The audit table shows no numeric, citation, or risk-label failures because every sentence cites the exact fact identifiers used by the formula. This example illustrates why paragraph-level review is insufficient: every claim requires a different evidence set and must be verified independently.

Figure 6 summarizes the verifier dimensions as pass rates. The pattern is consistent with the table-level results: No RAG cannot pass citation checks, Plain RAG passes many numeric checks but remains vulnerable to unsupported citations, and Verified RAG passes all numeric and citation checks in this run. The few residual verified-RAG errors are risk-support errors rather than arithmetic errors. They occur when the textual risk label is semantically close to the threshold boundary but the categorical label does not exactly match the mechanical score. This residual category is useful because it shows that verification must cover qualitative labels as well as numeric ratios.

The bootstrap confidence intervals in Table 8 show that the differences are stable at the claim level. Plain RAG has a 95% bootstrap interval of 25.55% to 31.81% for the claim-level error rate. Verified RAG has an interval of 0.14% to 1.11%. The gap remains large even though the memo generator and corpus are deterministic because the bootstrap resamples the audited claims rather than the generator seed. The empirical conclusion is therefore direct: on this dataset and task definition, retrieval alone reduces numerical hallucination but does not satisfy evidence-grounded financial reporting; verification is the component that changes the reliability regime.

Overall, the empirical results support a conservative deployment rule for financial memo generation. A RAG system should not be accepted merely because it retrieves relevant filings or because its prose includes citations. The system should be evaluated at the claim level, and every audited sentence should be connected to the formula and fact rows that produced it. Under that rule, No RAG fails by design, Plain RAG improves but remains unreliable for publication-quality numerical reporting, and Verified RAG satisfies the task-level audit for nearly all claims in the measured corpus.

Table 12. Claim templates and audit tolerances.

claim_type	generated_sentence	audit_tolerance	citation_requirement
revenue_yoy	Revenue changed by X year over year.	0.75 percentage points	current and prior revenue fact IDs
net_margin	Net margin was X.	0.50 percentage points	net income and revenue fact IDs

leverage	Liabilities-to-assets was X.	0.50 percentage points	liabilities and assets fact IDs
cfo_to_rev	Operating-cash-flow-to-revenue was X.	0.50 percentage points	CFO and revenue fact IDs
liquidity	Cash-to-liabilities was X.	0.50 percentage points	cash and liabilities fact IDs
risk_label	The computed financial risk label was X.	exact label match	all facts used by risk formula

Limitations

The experiment evaluates numerical and citation consistency, not legal sufficiency or investment merit. SEC-style financial data is an as-filed representation of registrant submissions, and the verifier checks consistency with the evidence base rather than the economic truth of the filing. A filing error or taxonomy inconsistency will propagate through the evidence base unless a separate external validation layer is added.

The evaluation corpus is intentionally controlled so that every audited claim has an observable answer. This improves internal validity but narrows the external scope. Real filings contain restatements, amendments, multiple segments, non-standard fiscal calendars, custom extensions, and cases where a risk statement depends on narrative context rather than a single numeric fact. The same verifier framework can be extended to those cases, but the current reported results should be read as a numerical-consistency benchmark rather than a complete production filing-analysis system.

The generation layer is deterministic and LLM-style rather than a proprietary or open-weight transformer executed through an API. This choice makes every failure reproducible and isolates the effect of retrieval and verification, but it does not measure model-family differences in fluency or reasoning. A direct model benchmark can plug into the same auditor by replacing the deterministic generator with an API or local model and keeping the claim parser and verifier unchanged.

The risk score is a transparent mechanical label, not a learned probability of default, distress, or market underperformance. Its purpose is to create a repeatable qualitative claim that still depends on several numerical facts. A production system would need expert-calibrated risk definitions, industry-specific thresholds, validation against historical outcomes, and governance review. The present experiment deliberately separates evidence-grounding accuracy from predictive financial modeling.

The corpus focuses on 2023 Q1–Q2 numerical statement facts and comparable prior-period values. It does not include MD&A narrative, risk-factor text, footnotes beyond the fact-level metadata, or market data. The conclusions therefore apply to numerical corporate risk memos whose claims can be expressed as formulas over XBRL facts. Qualitative risks such as litigation, supply-chain disruption, or regulatory exposure require additional evidence sources and separate support checks.

The replication package contains an offline SEC-format evaluation corpus plus a downloader for the official SEC Q1–Q2 2023 archives. In environments with direct internet access, the downloader can be used to replace or expand the local corpus with the full official archives. The reported numbers in this manuscript are the measured outputs of the included corpus and code.

Conclusion

This paper evaluated evidence-grounded financial RAG for corporate risk memo generation using SEC-format 2023 Q1–Q2 financial statement data. The empirical results show that plain retrieval improves over no retrieval but still leaves substantial numerical and citation errors. Verified RAG changes

the outcome by recomputing financial ratios, checking period-specific facts, and validating source identifiers before finalizing memo text. In the measured run, verified RAG reduced the claim-level error rate to 0.56%, removed numeric and citation errors, and produced a 96.67% memo-level all-claims pass rate.

The measured improvement is not a consequence of adding more prose context; it comes from changing the unit of control. Plain RAG retrieves context and asks the generator to reason over it. Verified RAG retrieves facts, computes derived values, and then permits only text whose numbers and citations match the computed audit record. This shift turns a memo from an unsupported narrative into a collection of evidence-linked claims. For corporate risk workflows, that distinction is essential because users must be able to inspect the origin of every percentage and every risk label.

The central lesson is architectural. Financial RAG should not treat a retrieved passage as sufficient evidence for a generated number. Each number in the memo must be connected to a fact identifier, every ratio must be recomputed, and every citation must cover the facts used by the formula. For trustworthy financial language systems, retrieval is the entry point; executable verification is the control that makes numerical claims auditable.

References

[1] U.S. Securities and Exchange Commission, "Financial Statement Data Sets," Washington, DC, USA, 2021.

[2] R. Debreceny, G. L. Gray, and A. Rahman, "The determinants of Internet financial reporting," J. Accounting Public Policy, vol. 21, no. 4–5, pp. 371–394, 2002.

[3] Xinzhuo Sun, Yifei Lu, and Jing Chen, “Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting”, JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[4] T. Loughran and B. McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," J. Finance, vol. 66, no. 1, pp. 35–65, 2011.

[5] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[6] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Inf. Process. Manage., vol. 24, no. 5, pp. 513–523, 1988.

[7] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333–389, 2009.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in Proc. ICLR Workshop, 2013.

[9] A. Vaswani et al., "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[10] 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.

[11] T. Brown et al., "Language models are few-shot learners," in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 1877–1901.

[12] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 9459–9474.

[13] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, 2020, pp. 6769–6781.

[14] Daren Zheng, Chenyu Li, and Harvey Davidson, “Continual Red-Teaming for In-the-Wild Jailbreaks via Online Guardrail Updates and Guardrail Distillation”, JACS, vol. 3, no. 2, pp. 35–49, Feb. 2023, doi: 10.69987/JACS.2023.30203.

[15] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to answer open-domain questions," in Proc. ACL, 2017, pp. 1870–1879.

[16] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in Proc. ACL, 2020, pp. 1906–1919.

[17] W. Kryściński, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," in Proc. EMNLP, 2020, pp. 9332–9346.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proc. KDD, 2016, pp. 1135–1144.

[19] R. Guidotti et al., "A survey of methods for explaining black box models," ACM Comput. Surv., vol. 51, no. 5, pp. 1–42, 2018.

[20] D. Dua et al., "DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs," in Proc. NAACL-HLT, 2019, pp. 2368–2378.

[21] A. Amini et al., "MathQA: Towards interpretable math word problem solving with operation-based formalisms," in Proc. NAACL-HLT, 2019, pp. 2357–2367.

[22] Siming Zhao, Hailin Zhou, and Daniel Martinez, “LLM-Assisted Causal Attribution of Service Performance Upgrades on Churn and Tenure: Full Evaluation on the IBM Telco Customer Churn Dataset”, JACS, vol. 3, no. 2, pp. 18–34, Feb. 2023, doi: 10.69987/JACS.2023.30202.

[23] D. Hendrycks et al., "Measuring mathematical problem solving with the MATH dataset," in Proc. NeurIPS, 2021, pp. 1626–1644.

[24] A. Jacovi and Y. Goldberg, "Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness?" in Proc. ACL, 2020, pp. 4198–4205.

[25] Yuanzheng Chen, Yitian Zhang, David Chau, and Matt Sherman, “Credit Card Default Risk Tiering with Probability Calibration and Uncertainty-Driven Rejection: A Reproducible Study on the UCI Credit Card Clients Dataset”, JACS, vol. 3, no. 4, pp. 31–47, Apr. 2023, doi: 10.69987/JACS.2023.30403.