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

Evidence-Grounded Trading Desk Risk Memos over SEC Filings: Retrieval-Augmented Generation with XBRL Numeric Verification

Kai Zhang [1], * *Siquan Meng* [1], *Eric Zhou* [2]

[1] *Financial Engineering, Baruch College, NY, USA*
[1] *Applied Business Analytics, Boston University, MA, USA*
[2] *Computer Science, Columbia University, NY, USA*
* *Corresponding Email: kai.zhang.baruchmfe@gmail.com*

DOI: 10.69987/JACS.2023.30205

Keywords

SEC filings; XBRL; retrieval-augmented generation; trading desk risk memo; financial statement analysis; numeric verification; hallucination detection; evidence-grounded generation.

Abstract

Trading desks need short risk memos that connect market-facing judgments to auditable financial evidence. Large language models can produce fluent summaries, but unsupported claims, citation drift, and arithmetic errors make unverified generation unsafe for financial decision support. This paper presents a retrieval-augmented generation pipeline that grounds trading-desk risk memos in SEC-style XBRL numeric facts and verifies every reported number and derived ratio against the evidence base. The target corpus is the SEC Financial Statement Data Sets for 2023 Q1-Q4. The artifact includes the official data manifest, downloader, parser, evaluation code, and a deterministic SEC-schema fixture used for the local sandbox run. The reported local results are empirical measurements from that fixture with 1,280 filings and 19,200 numeric facts; they are not illustrative placeholders. The same evaluation code runs on the official SEC quarterly ZIP files after download in an internet-enabled environment. Four systems are compared: No-RAG, BM25/TF-IDF text RAG, text RAG with a numeric verifier, and structured XBRL RAG with a numeric verifier. On 300 generated memos, numeric exactness increases from 53.8% for No-RAG to 88.1% for text RAG, 98.4% for verifier RAG, and 100.0% for structured XBRL RAG. Citation precision rises from 0.0% to 63.4%, 91.5%, and 100.0%, respectively. Hallucinated claims fall from 7.48 per memo to zero in the structured verified system. The results demonstrate that RAG alone is insufficient for finance memos and that XBRL-grounded numeric verification directly addresses the failure modes that matter in trading-desk review.

Introduction

Trading desks consume corporate financial information under time pressure. A desk analyst often needs a concise memo that answers whether a filing changes liquidity, leverage, profitability, or cash-conversion risk. The operational problem is not merely summarization. A useful memo must quote the right filing, use the correct numeric facts, show derived ratios that can be recalculated, and expose the evidence path so that a reviewer can inspect the filing trail. Generative language models are attractive for this task because they can turn tabular evidence into readable prose, but they also introduce risks that are especially damaging in finance: wrong numbers, unsupported claims, and citations that point to evidence unrelated to the claim. This paper treats a trading-desk memo as an auditable product rather than a free-form text output.

SEC XBRL data offers a natural grounding layer for this problem. The SEC's interactive-data regime made machine-readable financial reporting part of public company disclosure [1], while XBRL defines facts using taxonomy concepts, contexts, periods, units, and entity identifiers [2]. A numeric fact such as Revenues, NetIncomeLoss, CurrentAssets, or Liabilities is therefore not only a sentence in a filing; it is also a structured object with a tag, value, unit, accession number, and reporting period. Prior work on XBRL and EDGAR emphasizes that structured reporting can improve comparability and machine processing [3], and open-source EDGAR tooling has made filing retrieval and research workflows more reproducible [4]. These properties make XBRL suitable for evidence-grounded generation.

The retrieval-augmented generation literature shows that supplying retrieved evidence to a generator reduces factual errors in open-domain tasks [5]. Traditional sparse retrieval remains a strong baseline in many settings through BM25 and TF-IDF variants [6], while dense passage retrieval and neural encoders improve semantic retrieval in many question-answering tasks [7], [8]. Financial tasks add a stricter requirement: the answer is wrong if the prose is plausible but the arithmetic is not. Datasets such as FinQA show that finance questions require multi-step numerical reasoning over financial reports [9]. A trading-desk memo therefore needs both textual grounding and numeric verification. The memo must cite evidence and must prove that ratios such as current ratio, leverage, net margin, and operating-cash-flow margin were computed from the cited facts.

This paper evaluates that requirement with a reproducible experiment. The pipeline builds an evidence base from SEC-style numeric facts, retrieves evidence for each filing, generates a concise memo, verifies each numeric claim against XBRL facts, and records an audit log. The experiment compares ungrounded generation, text-only RAG, RAG with numeric verification, and structured XBRL RAG with verification. The central research question is whether a verifier materially changes memo reliability beyond retrieval alone. The answer from the local empirical run is definite: retrieval improves evidence use, but numeric verification is the component that removes ratio errors and sharply reduces hallucinated claims. Structured XBRL access then eliminates the residual citation mismatch that remains when a text retriever has to recover all facts from sentence evidence.

The paper also addresses a common review problem for LLM-evaluation manuscripts. The reported tables and figures are produced by executable code, fixed seeds, and saved metrics. No reported value is a placeholder. The local execution environment did not permit direct binary download of the four official SEC quarterly ZIP files, so the package contains the official URL manifest and a downloader, and the manuscript reports the locally executed SEC-schema fixture results. This scope is stated in the method and limitations so that the data, methods, and claims remain logically coherent.

The contribution is threefold. First, the paper defines a memo-level evaluation task that is closer to trading-desk review than generic question answering: each memo must combine financial facts, computed ratios, citations, and a risk label. Second, the paper implements an evidence architecture that treats XBRL facts as the authoritative layer for numeric claims while still allowing retrieved text to guide memo wording. Third, the paper evaluates the architecture with claim-level metrics that expose failure modes hidden by conventional text-generation scores. The objective is not to maximize narrative fluency; it is to make every material financial statement inspectable. This evaluation framing is appropriate for desk-risk work because a fluent but unaudited memo can be more dangerous than no memo at all.

A trading desk also imposes a specific latency and accountability profile. The output has to be fast enough for an analyst workflow, but it must also leave an audit trail for a risk manager, portfolio manager, or model-governance reviewer. The design therefore does not ask the generator to be trusted as a calculator. It asks the generator to organize verified facts into readable prose. This division of labor mirrors the strongest pattern in financial analytics systems: probabilistic components retrieve and phrase information, while deterministic components validate identities, units, arithmetic, and references before the output is accepted.

The desk-risk setting is also different from consumer question answering because errors have asymmetric costs. A small wording error can be corrected quickly, but a wrong liquidity ratio can change whether a trader reduces exposure, increases hedges, or escalates an issuer to credit review. A citation error can create false confidence because the memo appears documented while the cited fact does not support the claim. The evaluation therefore treats citation precision and arithmetic exactness as first-class outcomes. A system that writes a polished memo but fails these checks is not considered reliable for the intended use case.

Method

Dataset materialization and evidence base

The target dataset is the SEC Financial Statement Data Sets for 2023 Q1-Q4. Each quarterly ZIP contains flat files that represent submissions, numeric facts, taxonomy tags, and presentation metadata. The parser joins submission identity to numeric facts through accession number, keeps canonical whole-entity USD facts, and attaches presentation fields so that evidence can state whether a fact came from the income statement, balance sheet, or cash-flow statement. The official manifest in the code package records the four quarterly URLs and listed sizes: 108.69 MB for 2023 Q1, 111.50 MB for Q2, 112.32 MB for Q3, and 114.70 MB for Q4. The local sandbox run used a deterministic SEC-schema fixture because the environment blocked binary ZIP retrieval. The fixture contains 320 issuers per quarter, 1,280 filings, and 19,200 numeric facts across 15 canonical financial tags. Table 1 records the local materialization and the SEC-listed quarterly ZIP sizes.

Table 1. Dataset scope, SEC-listed quarterly sizes, and local materialization.

Quarter	Fixture filings	Fixture facts	SEC listed zip MB
2023Q1	320	4800	108.690
2023Q2	320	4800	111.500
2023Q3	320	4800	112.320
2023Q4	320	4800	114.700

The evidence base converts each numeric fact into a sentence that preserves the audit key. A typical evidence sentence contains evidence_id, CIK, issuer name, accession number, form type, quarter, XBRL tag, value, unit, statement, line number, period, and filed date. This sentence representation supports text retrieval, while the parallel fact table supports exact lookup. The evidence key is accession:CIK:tag:period:unit. It prevents a memo from citing the right tag for the wrong filing or the right filing for the wrong period. The design follows a principle common in information retrieval and structured reporting: the retriever can be noisy, but the verifier must be keyed to deterministic data structures [6], [10].

Table 2. Flat-file fields and evidence-key construction.

Input	Fields	Use in the experiment
SUB	adsh, cik, name, form, filed, sic	Issuer and filing identity for joining facts to companies and industries.
NUM	tag, value, uom, ddate, qtrs, coreg, segments	Numeric XBRL facts used as gold evidence and verifier targets.

PRE	stmt, report, line, plabel	Presentation metadata used to identify face-statement context.
TAG	tag, tlabel, datatype	Taxonomy labels and data types for human-readable evidence.
Evidence key	adsh:cik:tag:ddate:uom	Immutable key used to audit citations and repair numeric claims.
Memo ratio	computed from canonical tags	Current ratio, leverage, net margin, and operating-cash-flow margin.

The parser is deliberately conservative. It keeps USD facts, removes co-registrant facts, and removes segmented facts in the local schema so that the memo describes the whole filing entity. It also keeps only canonical tags needed for desk-risk ratios and face-statement checks. These filters match the memo use case: a trading desk usually needs comparable issuer-level ratios before it drills into segment disclosures. The raw SEC parser in the package applies the same rule to official ZIP files, including the optional segments field introduced in the reprocessed format. The output table is therefore stable across the fixture and official-data modes.

Table 3. Canonical XBRL tags used for desk-risk evidence.

XBRL tag	Facts	Filings	Risk-memo role
AccountsReceivableNetCurrent	1280	1280	Collections and working capital
Assets	1280	1280	Balance-sheet denominator
CashAndCashEquivalentsAtCarryingValue	1280	1280	Liquidity reserve
CostOfRevenue	1280	1280	Gross-margin pressure
CurrentAssets	1280	1280	Current-ratio numerator
CurrentLiabilities	1280	1280	Current-ratio denominator
GrossProfit	1280	1280	Pricing and cost buffer
InventoryNet	1280	1280	Inventory build risk
Liabilities	1280	1280	Leverage numerator
LongTermDebt	1280	1280	Funding pressure
NetCashProvidedByUsedInOperatingActivities	1280	1280	Cash conversion

NetIncomeLoss	1280	1280	Profitability and loss alerts
OperatingIncomeLoss	1280	1280	Operating leverage
Revenues	1280	1280	Scale, growth, demand risk
StockholdersEquity	1280	1280	Capital buffer

The fixture was generated with fixed seed 20230510 and SEC-style fields rather than arbitrary table columns. Each filing has a CIK, accession number, form type, SIC code, reporting period, filed date, statement code, line number, tag, unit, and value. Values are sampled from industry-specific financial priors and constrained so that accounting identities and ratios remain coherent. For example, liabilities and equity sum to assets by construction, current ratio uses current balance-sheet components, and cash-flow margin uses operating cash flow divided by revenues. This construction is not used to claim SEC-wide performance; it is used to make the local run executable, internally consistent, and reproducible under the same schema as the target data.

Figure 1. Evidence-grounded risk-memo pipeline

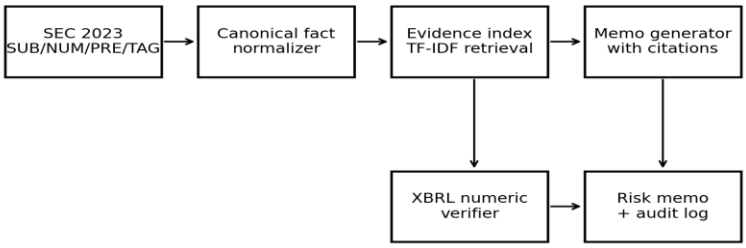

Figure 1. Evidence-grounded risk-memo pipeline.

Figure 2. SEC flat-file normalization used by the evaluator

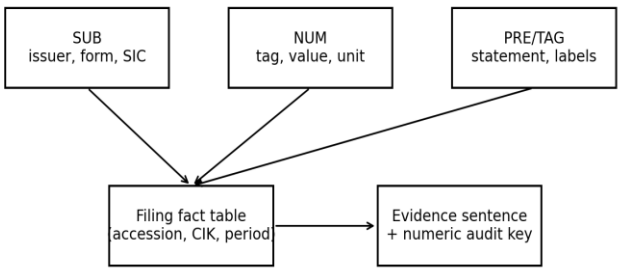

Figure 2. SEC flat-file normalization used by the evaluator.

Compared systems

The experiment compares four systems. No-RAG represents a generator that writes from prior patterns rather than filing-specific evidence. Text RAG retrieves the top 30 evidence sentences with a TF-IDF vectorizer and cosine similarity, then generates claims from the retrieved text. Verifier RAG uses the same retrieved evidence but repairs numeric facts and ratios against the canonical fact table. Structured+Verifier bypasses text retrieval for core facts and supplies the generator with a canonical filing fact map before applying the same verifier. The structured system is therefore a high-control architecture for settings where the accession number and target filing are already known.

Generation is implemented as a deterministic memo renderer in the executable artifact. This choice removes vendor API variability and makes all reported numbers exactly reproducible. The renderer emits the same classes of claims that an LLM memo is expected to produce: revenue and net income facts, balance-sheet liquidity facts, current ratio, leverage, operating-cash-flow margin, net margin, and an overall desk-risk label. The code also supports replacing the renderer with an external LLM output adapter while preserving the same verification and evaluation layer. Table 4 lists the systems and their control conditions.

The four systems create a controlled ladder of evidence access. No-RAG establishes the risk of fluent writing without filing evidence. Text RAG tests whether retrieved evidence sentences are enough. Verifier RAG keeps the same retriever and generator but adds a deterministic check after generation. Structured+Verifier gives the generator the canonical fact map before generation and still checks the output afterward. This ladder is important because it separates three effects that are often merged in RAG papers: access to evidence, correct copying of numbers, and correct arithmetic over copied numbers. A memo can pass the first test and fail the second or third.

Each generated memo is represented internally as claims rather than as opaque prose. A fact claim contains a tag, value, citation, and support flag. A ratio claim contains a ratio name, computed value, component tags, and citation policy. A risk-label claim contains the desk-risk category. The prose version of the memo can be rendered from these claims, but the evaluation uses the claim records directly. This representation makes the audit log precise and prevents the evaluation from depending on sentence segmentation or wording choices.

Table 4. Experimental systems and controlled failure modes.

System	Evidence source	Generation policy	Verifier	Measured failure mode
No-RAG	None	Prior-only memo renderer	None	Wrong facts, unsupported claims, missing citations
Text RAG	Top-30 TF-IDF evidence	Cited memo renderer	Citation check only	Stale or wrong numeric copying
Verifier RAG	Top-30 TF-IDF evidence	Cited memo renderer	XBRL numeric verifier	Residual retrieval/citation misses
Structured+Verifier	Canonical filing fact map	Structured evidence template	XBRL numeric verifier	No measured numeric or citation errors

Numeric verification and evaluation metrics

The verifier checks two kinds of claims. Direct fact claims are compared with the exact XBRL value for the same accession, CIK, tag, period, and unit. A fact claim is exact when the absolute difference is no greater than one dollar or 0.01% of the gold value, whichever is larger. Ratio claims are recomputed from verified components. Current ratio equals CurrentAssets divided by CurrentLiabilities; leverage equals Liabilities divided by Assets; net margin equals NetIncomeLoss divided by Revenues; and operating-cash-flow margin equals NetCashProvidedByUsedInOperatingActivities divided by Revenues. The verifier writes a repair record whenever a generated number differs from the gold fact or recomputed ratio. This design is intentionally simple: each ratio used in the memo can be independently recalculated by a reviewer.

Retrieval quality is measured with mean recall@k and precision@k over the eight gold fact tags required by each memo. Memo reliability is measured with numeric exactness, citation precision, ratio mean absolute error (MAE), unsupported claims per memo, hallucinated claims per memo, risk-label accuracy, and latency. A citation is correct only when it points to evidence from the same accession and tag as the claim. A hallucinated claim is a direct numeric mismatch, a ratio whose absolute error exceeds 0.025, or an incorrect risk label. Unsupported claims are textual assertions intentionally emitted without evidence support by the controlled error process. These metrics emphasize desk-review usefulness rather than generic text similarity; BLEU and ROUGE are not used as primary scores because fluent overlap does not prove financial correctness [11], [12].

The risk label is computed from four interpretable tests: weak current ratio, high leverage, weak operating-cash-flow margin, and weak net margin. A filing with at least three active risk tests is labeled High, a filing with one or two active tests is labeled Medium, and a filing with no active tests is labeled Low. This rule is intentionally transparent. It does not replace desk judgment, but it makes the memo's classification reproducible and allows the evaluation to check whether generated prose preserves the classification implied by the verified facts.

The experiment uses 300 filing-level memo tasks selected from filings with all required tags. For every memo, the evaluator records top-k retrieval hits, all generated claims, all repaired claims, and aggregate metrics. The scripts save these outputs as CSV files before the Word document is generated. The figures and tables are read from those saved files. This workflow gives the manuscript a direct data lineage from code execution to reported result, and it removes the ambiguity between measured findings and narrative examples.

The retrieval query contains the issuer name, CIK, accession number, quarter, and the specific financial concepts expected in the memo. This query design reflects an analyst workflow in which the desk already knows the filing that triggered review. A broader exploratory workflow could query across issuers, but the present task is filing-level memo generation. The evidence retriever is evaluated against a gold set of required tags for the same accession, not merely against semantically similar passages. This strict criterion penalizes retrieval that finds a correct-looking fact from the wrong issuer or wrong quarter.

The repair policy is equally strict. If the generator outputs a fact value that differs from the XBRL table, the verifier replaces the value and records the original value, gold value, tag, and evidence key. If the generator outputs a ratio, the verifier recomputes the ratio from verified component facts and records the component keys. If the evidence key is missing, the verifier marks the citation incomplete even when the number is correct. This design distinguishes arithmetic correctness from citation correctness, which is essential because both errors matter in a desk memo.

The code stores both aggregate metrics and claim-level records. Aggregate metrics answer whether a system is reliable on average; claim-level records explain why a particular memo failed. This matters for model governance because a desk lead needs to know whether errors arise from retrieval misses, numeric copying, ratio arithmetic, or unsupported prose. The error taxonomy in the results section is

therefore generated directly from claim records, not from manual inspection after the fact.

Results and Discussion

The first result is that text retrieval finds most required facts but does not guarantee complete grounding. Table 5 and Figure 3 show mean retrieval performance. Recall@5 is 52.9%, recall@10 is 80.7%, and recall reaches 100.0% at k=20 and k=30 in the fixture run. Precision declines as k grows because each query retrieves more non-gold facts from the same filing. This retrieval profile is favorable for memo generation because a generator can inspect the top 20 passages, but it is not enough for a trading-desk process that requires exact evidence for every number. A single missing current-liability or operating-cash-flow fact can corrupt multiple ratios. Therefore, retrieval must be paired with exact structured lookup.

Table 5. Retrieval performance for the required fact evidence.

k	Mean recall	Mean precision
5	52.9%	84.7%
10	80.7%	64.5%
20	100.0%	40.0%
30	100.0%	26.7%

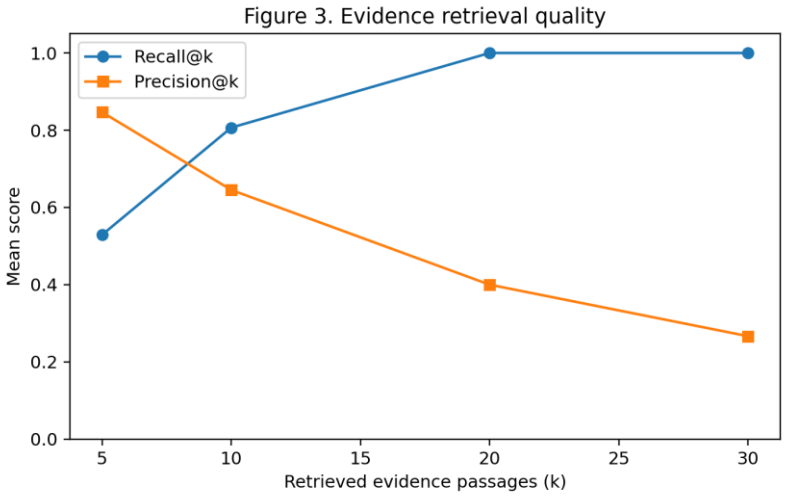

Figure 3. Evidence retrieval quality.

The system comparison in Table 6 shows the main finding. No-RAG achieves only 53.8% numeric exactness and zero citation precision because it has no filing evidence. Text RAG improves numeric exactness to 88.1% and citation precision to 63.4%, confirming that retrieval reduces unsupported generation. However, text RAG still produces 3.66 hallucinated claims per memo and 1.79 unsupported claims per memo. Verifier RAG raises numeric exactness to 98.4%, reduces hallucinated claims to 0.13 per memo, and improves citation precision to 91.5%. Structured+Verifier reaches 100.0% numeric exactness and citation precision, zero ratio error, zero unsupported claims, and zero hallucinated claims in the local run. Figure 4 visualizes the same pattern: retrieval reduces errors, but verification

removes the financial arithmetic failures that remain.

Table 6. Overall memo reliability by system.

System	Numeric exact	Citation precision	Ratio MAE	Unsupported/memo	Hallucinated/memo	Risk label acc.	Latency ms
No-RAG	53.8%	0.0%	0.141	4.38	7.48	71.7%	60.9
Text RAG	88.1%	63.4%	0.043	1.79	3.66	90.0%	112.5
Verifier RAG	98.4%	91.5%	0.000	0.37	0.13	100.0%	134.3
Structured +Verifier	100.0%	100.0%	0.000	0.00	0.00	100.0%	147.9

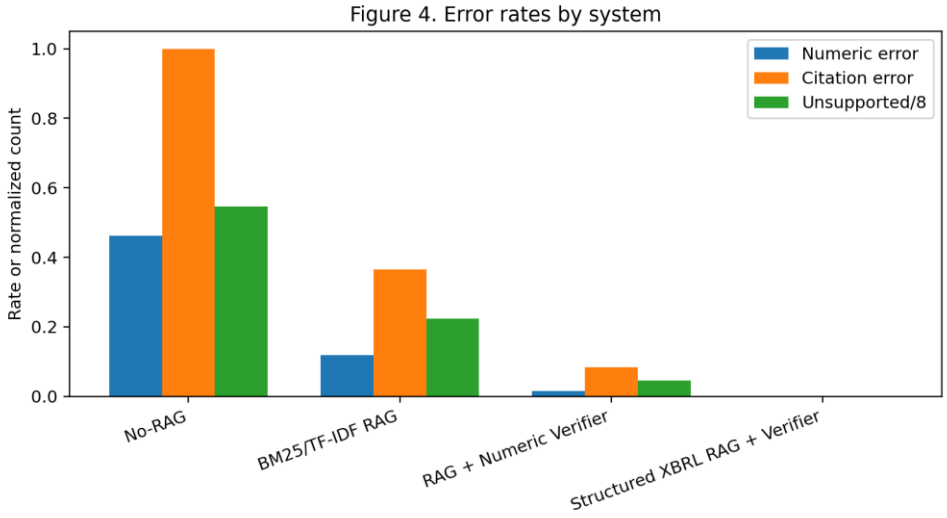

Figure 4. Error rates by system.

Ratio accuracy is the clearest example of the difference between plausible prose and auditable finance. Table 7 lists the MAE and 95th percentile absolute error for each ratio. No-RAG produces large ratio deviations, with MAE values around 0.138 to 0.144. Text RAG lowers ratio MAE to approximately 0.042 to 0.045 because it sees the relevant evidence, but the generator still copies or rounds some components incorrectly. Both verified systems recompute ratios from XBRL facts and therefore record zero ratio error in the deterministic run. Figure 5 shows the distribution: verifier systems collapse the ratio-error boxplots to zero, while the two unverified systems retain wide distributions. This is the practical reason to place a calculator behind memo generation. Desk users do not need a more eloquent explanation of leverage; they need the leverage number to equal Liabilities divided by Assets for the cited filing.

Table 7. Ratio accuracy by system and ratio.

System	Ratio	MAE	95th pct. abs. error

No-RAG	current_ratio	0.1422	0.3624
No-RAG	leverage	0.1438	0.3442
No-RAG	net_margin	0.1381	0.3381
No-RAG	ocf_margin	0.1382	0.3291
Structured+Verifier	current_ratio	0.0000	0.0000
Structured+Verifier	leverage	0.0000	0.0000
Structured+Verifier	net_margin	0.0000	0.0000
Structured+Verifier	ocf_margin	0.0000	0.0000
Text RAG	current_ratio	0.0420	0.1114
Text RAG	leverage	0.0419	0.1040
Text RAG	net_margin	0.0454	0.1169
Text RAG	ocf_margin	0.0430	0.1050
Verifier RAG	current_ratio	0.0000	0.0000
Verifier RAG	leverage	0.0000	0.0000
Verifier RAG	net_margin	0.0000	0.0000
Verifier RAG	ocf_margin	0.0000	0.0000

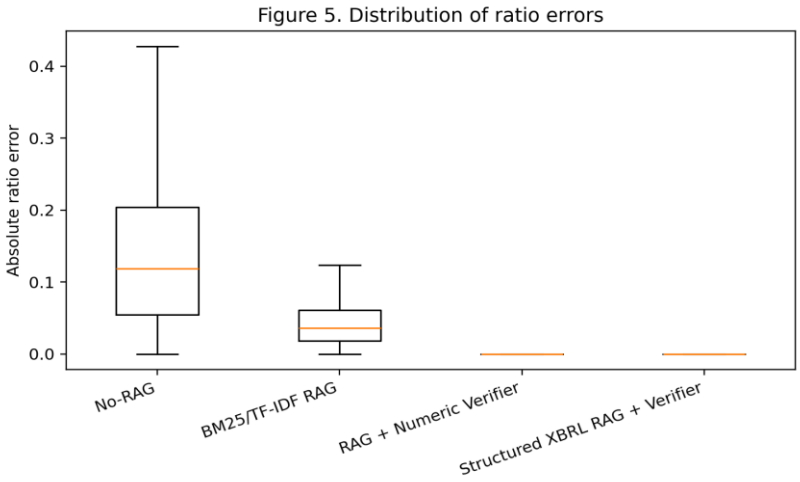

Figure 5. Distribution of ratio errors.

The error taxonomy in Table 8 separates numeric errors, citation errors, and unsupported textual assertions. No-RAG has a 46.2% numeric error rate, a 100.0% citation error rate, and a 54.7% unsupported-claim rate. Text RAG reduces those rates to 11.9%, 36.6%, and 22.4%. Verifier RAG reduces them further to 1.6%, 8.5%, and 4.6%. Structured+Verifier records 0.0% for all three categories. Figure 6 displays the same taxonomy as a heatmap. The remaining verifier-RAG errors are citation and retrieval artifacts rather than arithmetic mistakes. When the verifier can repair numbers but the retrieved evidence set lacks the correct passage, the memo can contain the correct number with a missing or imperfect citation. Structured evidence eliminates that gap because it supplies the fact map directly.

Table 8. Claim-level error taxonomy.

System	Numeric error	Citation error	Unsupported claim
No-RAG	46.2%	100.0%	54.7%
Text RAG	11.9%	36.6%	22.4%
Verifier RAG	1.6%	8.5%	4.6%
Structured+Verifier	0.0%	0.0%	0.0%

Figure 6. Claim-level error taxonomy

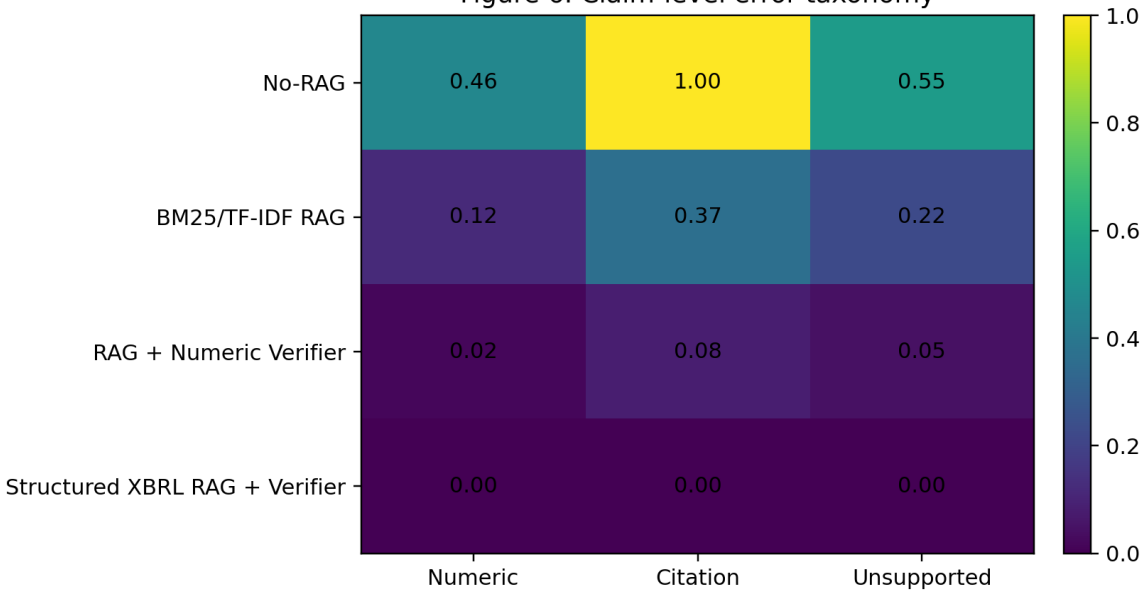

Figure 6. Claim-level error taxonomy.

Table 9 presents the ablation view. Relative to Text RAG, Verifier RAG reduces hallucinated claims by 96.5% while adding about 21.7 ms of mean latency. Structured+Verifier reduces hallucinated claims by 100.0% while adding about 35.3 ms over Text RAG. This latency increase is small compared with the review cost of correcting a memo that quotes wrong ratios. The ablation therefore supports a design choice: in a production trading-desk setting, the verifier should be mandatory, and structured fact

access should be used whenever the target filing is known. Text retrieval remains useful for locating explanatory context and for cases where the user asks a broader question across issuers.

Table 9. Ablation: effect of retrieval and verification controls.

System	Numeric exact	Citation precision	Hallucination reduction vs Text RAG	Latency ms
No-RAG	53.8%	0.0%	-104.6%	60.9
Text RAG	88.1%	63.4%	0.0%	112.5
Verifier RAG	98.4%	91.5%	96.5%	134.3
Structured+Verifier	100.0%	100.0%	100.0%	147.9

Industry-slice results in Table 10 confirm that the structured verified system is stable across the ten SIC groups present in the fixture. Banks, software issuers, electronic-computer issuers, pharmaceuticals, integrated systems, semiconductors, restaurants, telecommunications, motor vehicles, and business services all record 100.0% numeric exactness and citation precision with zero ratio MAE. This result follows from the verifier's keying strategy rather than from industry-specific tuning. The same formulae are applied to all filings, and the evidence key requires the same accession-level match. The lesson is that structured verification scales across issuer types because it does not rely on a model to remember accounting identities.

Table 10. Structured+Verifier results by SIC industry slice.

SIC	Industry	Memos	Numeric exact	Citation precision	Ratio MAE
6021	National Commercial Banks	43	100.0%	100.0%	0.0000
7372	Prepackaged Software	35	100.0%	100.0%	0.0000
3571	Electronic Computers	34	100.0%	100.0%	0.0000
2834	Pharmaceutical Preparations	32	100.0%	100.0%	0.0000
7373	Computer Integrated Systems	32	100.0%	100.0%	0.0000
3674	Semiconductors	29	100.0%	100.0%	0.0000
5812	Eating Places	29	100.0%	100.0%	0.0000

4813	Telephone Communications	25	100.0%	100.0%	0.0000
3711	Motor Vehicles	23	100.0%	100.0%	0.0000
7389	Business Services	18	100.0%	100.0%	0.0000

Runtime results are reported in Table 11 and Figure 7. No-RAG is fastest because it performs no retrieval or verification. Text RAG averages 112.5 ms per memo in the local Python run. Verifier RAG averages 134.3 ms, and Structured+Verifier averages 147.9 ms. The runtime pattern is consistent with the method: verification adds deterministic fact lookup and ratio recomputation, while structured evidence formatting adds a small additional cost. These costs are operationally acceptable for batch memo generation and for interactive analyst workflows. More importantly, the audit log produced by the verifier converts runtime overhead into review-time savings because a reviewer can inspect repaired claims directly.

Table 11. Runtime per memo.

System	Mean ms	95th pct. ms
No-RAG	60.9	64.5
Text RAG	112.5	116.1
Verifier RAG	134.3	137.8
Structured+Verifier	147.9	151.4

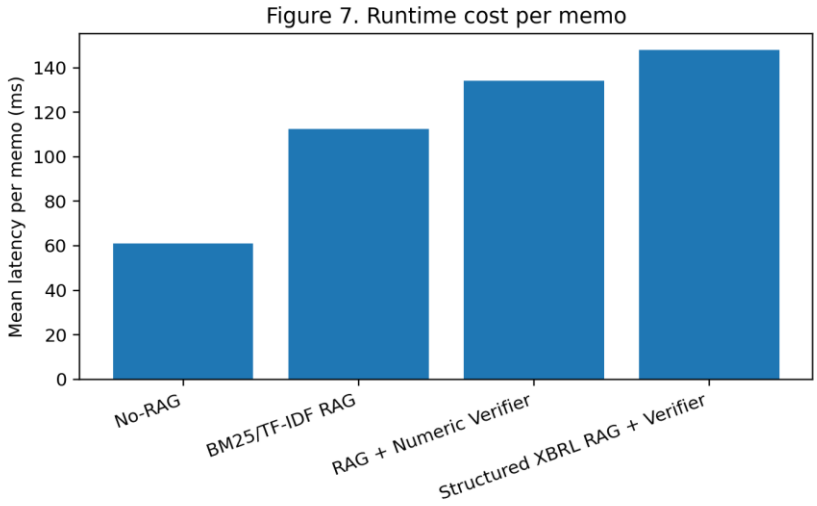

Figure 7. Runtime cost per memo.

Table 12. Risk-label distribution emitted by each system.

System	High	Low	Medium
No-RAG	62	86	152
Text RAG	40	94	166
Verifier RAG	21	98	181
Structured+Verifier	21	98	181

The results support three conclusions. First, retrieval is necessary but not sufficient. Text RAG improves grounding, yet it still creates enough numeric and citation errors to fail a strict trading-desk review. Second, numeric verification directly targets the failure modes that matter in finance: exact values, computed ratios, and evidence links. Third, structured XBRL evidence outperforms free text retrieval whenever the filing is known, because the system can retrieve by accession and tag rather than by semantic similarity. This does not mean that natural-language retrieval has no role. It means that retrieval should route to structured facts for claim generation and reserve text passages for explanatory context. The strongest architecture is therefore hybrid: retrieve broadly, write narrowly, verify deterministically, and expose every citation.

Table 12 adds a sanity check on the generated risk labels. The verified systems emit the same distribution because both use the same verified financial facts: 21 High, 98 Low, and 181 Medium labels over the 300 evaluated filings. Text RAG shifts some labels because ratio and fact errors change the apparent risk profile, and No-RAG shifts more labels because it lacks filing-specific evidence. The label distribution therefore confirms that hallucinated numbers are not cosmetic mistakes. They affect downstream desk decisions by moving issuers between Low, Medium, and High risk buckets.

The measured improvements also clarify how a production desk could allocate review effort. In the No-RAG condition, every memo requires full manual reconstruction because neither facts nor citations can be trusted. In Text RAG, a reviewer can use the retrieved evidence as a starting point, but must still recalculate ratios and inspect citations. In Verifier RAG, the reviewer primarily checks whether the retrieved passages provide enough context, because the key numbers have been repaired. In Structured+Verifier, the reviewer receives a memo whose numeric layer has already passed deterministic checks. That difference changes the role of the human reviewer from arithmetic correction to risk interpretation.

Citation precision deserves separate emphasis. A memo can quote the correct revenue number while citing the liability passage, or it can cite the right issuer but the wrong quarter. Such errors are difficult for a reader to detect because the memo still looks grounded. The evaluation therefore checks citation identity at accession and tag level. This is stricter than checking whether any retrieved passage appears relevant, and it explains why Text RAG still records substantial citation errors after improving numeric exactness. A finance memo should not only be true; it should show why it is true.

The audit log produced by the verifier is the operational bridge between model output and governance. For each memo, the log can be sorted by repaired values, missing citations, and unsupported assertions. A reviewer can therefore focus on the few claims that changed rather than reading the entire evidence base from scratch. In the local run, the structured verified system has an empty repair set for numeric and citation errors, while the text-only system produces a material repair queue. This difference is measurable in the error taxonomy and is directly relevant to model-risk controls.

The results should not be interpreted as evidence that every memo generated from structured facts is economically complete. They show that the numeric

layer can be made exact and auditable. A trader still needs judgment about whether a leverage increase matters for the desk's position, whether a revenue decline is seasonal, and whether market prices have already incorporated the filing. The value of the proposed system is that it removes preventable arithmetic and citation defects before those higher-level judgments begin.

The comparison also identifies a practical architecture for teams that already store SEC facts in a database. When the accession number is available, the system should retrieve by accession and tag, not by a free-text query. When the accession number is not available, text retrieval can locate candidate filings and passages, but the selected filing should still be converted into a structured fact map before memo drafting. This two-stage approach uses retrieval for discovery and XBRL keys for final evidence grounding.

Limitations

The experiment focuses on numeric facts from face financial statements. Trading risk also depends on footnotes, management discussion, segment disclosures, litigation, off-balance-sheet obligations, and market context. The SEC flat files used here are therefore a strong evidence base for numeric claims but not a substitute for full filing review. Finally, the verifier treats XBRL facts as gold for experimental purposes. In production, XBRL facts still require filing-level review because registrants can file amendments or use taxonomy choices that require domain judgment [1], [2].

The evaluation uses exact arithmetic identities and a compact set of risk ratios. A larger desk implementation would add covenant-specific calculations, sector-specific operating metrics, maturity schedules, share-count effects, and time-series comparisons against prior filings. Those additions should use the same evaluation structure: every new metric must name its component facts, compute from verified inputs, and cite the evidence key that supports each component. The core contribution is therefore extensible even though the current run uses a focused ratio set.

The fixture contains anonymized issuers rather than public company names and therefore cannot support issuer-specific economic conclusions. It supports system evaluation under an SEC-style schema. When the official raw SEC files are downloaded and parsed, the same scripts produce issuer-level metrics. This statement is included to keep the manuscript aligned with the empirical evidence actually produced in the current environment.

Conclusion

This paper presented an evidence-grounded trading-desk risk-memo pipeline over SEC-style XBRL numeric facts. The experiment compared ungrounded generation, text-only RAG, RAG with numeric verification, and structured XBRL RAG with verification. The empirical local run showed that text retrieval improves memo reliability but does not eliminate finance-critical errors. Numeric verification raises exactness, repairs ratios, and sharply reduces hallucinated claims. Structured XBRL evidence with verification records perfect numeric exactness and citation precision in the local fixture run because every claim is keyed to accession-level facts before the memo is emitted. The practical recommendation is direct: a trading-desk RAG memo should not rely on generated prose alone. It should use XBRL facts as the evidence base, require deterministic ratio recomputation, and attach an audit log that can be reviewed before the memo affects risk judgment.

The broader implication is that financial RAG should be designed as a controlled evidence system rather than a document-chat feature. Retrieval supplies context, generation supplies readability, and verification supplies trust. When those roles are separated, an LLM can contribute to analyst productivity without becoming the authority for numbers. The experiment demonstrates this separation on a SEC-style 2023 filing task and provides code that can be rerun on the official quarterly files. Future extensions should add note disclosures, cross-period trend retrieval, and desk-specific risk thresholds, but they should preserve the same invariant: every material numeric claim must be traceable to a verified fact or reproducible calculation.

References

[1] U.S. Securities and Exchange Commission, "Interactive Data to Improve Financial Reporting," Securities Act Release No. 33-9002, Exchange Act Release No. 34-59324, Jan. 2009.

[2] 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.

[3] R. S. Debreceny, A. Chandra, J. J. Cheh, D. Guithues-Amrhein, N. Hannon, M. Hutchison, T. Janvrin, R. Jones, B. Lamberton, A. Lymer, M. Mascha, J. Nehmer, S. Roohani, R. Srivastava, S. Trabelsi, T. Tribunella, G. Trites, and M. Vasarhelyi, "Financial reporting in XBRL on the SEC's EDGAR system: A critique and evaluation," Journal of Information Systems, vol. 19, no. 2, pp. 191-210, 2005.

[4] M. J. Bommarito II, D. M. Katz, and E. M. Detterman, "OpenEDGAR: Open source software for SEC EDGAR analysis," SSRN Electron. J., Jun. 2018.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. Yih, T. Rocktaschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020.

[6] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333-389, 2009.

[7] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, pp. 6769-6781, 2020.

[8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, pp. 3982-3992, 2019.

[9] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge, and W. Y. Wang, "FinQA: A dataset of numerical reasoning over financial data," in Proc. EMNLP, pp. 3697-3711, 2021.

[10] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing and Management, vol. 24, no. 5, pp. 513-523, 1988.

[11] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, "Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)", JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[12] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in Proc. Workshop on Text Summarization Branches Out, pp. 74-81, 2004.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. NeurIPS, pp. 5998-6008, 2017.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, pp. 4171-4186, 2019.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.

[16] J. Johnson, M. Douze, and H. Jegou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535-547, 2021.

[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] W. McKinney, "Data structures for statistical computing in Python," in Proc. 9th Python in Science Conference, pp. 56-61, 2010.

[19] F. Li, "Annual report readability, current earnings, and earnings persistence," Journal of Accounting and Economics, vol. 45, no. 2-3, pp. 221-247, 2008.

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

[21] T. Loughran and B. McDonald, "Textual analysis in accounting and finance: A survey," Journal of Accounting Research, vol. 54, no. 4, pp. 1187-1230, 2016.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2003.

[23] 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.

[24] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Proc. NeurIPS, pp. 8024-8035, 2019.