EDGAR Analyst Copilot

Sangyeon Lee, Ronghao Zeng, Mingchen Yuan, Yuan Zhuang, Jinyu Li

Problem statement

Public companies are required to submit annual (10-K) and quarterly (10-Q) reports to the U.S. Securities and Exchange Commission (SEC). These filings contain extensive financial data and narrative disclosures that are critical for investment analysis. However, they are often hundreds of pages long, vary widely in format and structure across firms, and require substantial manual effort to interpret.

Existing AI tools, such as ChatGPT, are not purpose-built for financial reporting. They tend to generate unsupported information, misread numerical tables (e.g., units such as “in millions”), and fail to provide verifiable sources.

Our project builds a focused analytical assistant that can read complex financial filings, provide answers with citations, structured KPI tables, and clearly summarize shifts in risk factors to support faster and more consistent financial analysis.

Proposed Solution

We’re building an EDGAR Analyst Copilot — a smart assistant that helps analysts quickly make sense of 10-K and 10-Q filings. It can answer questions with inline citations, then pull out and standardize key financial KPIs into clear, comparable tables. And finally it summarizes risk factors while highlighting what’s new or changing. Under the hood, it uses retrieval-augmented generation (RAG), lightweight fine-tuning for finance-specific Q&A, and numeric guardrails to make sure every answer is accurate and backed by the source text.

Value Beyond Off-the-Shelf Solutions

Our system is not just another generic chatbot that reads filings, it is a purpose-built financial assistant specifically designed for SEC 10-K and 10-Q analysis. It introduces several key capabilities that go beyond standard LLM APIs:

Finance-Aware Retrieval: The retrieval process is customized for financial report structures, using section-aware chunking (e.g., MD&A, Risk Factors, Notes to Financials). Query relevance is weighted toward sections containing financial or risk content, rather than uniform vector matching.

Table-Grounded Numerical Reasoning: Automatically detects numerical units (e.g., “in millions”) and currency codes to ensure accurate scaling and normalization. Supports ratio derivation such as gross margin, liquidity ratio, and free-cash-flow margin instead of merely echoing raw numbers.

Strict Citation Enforcement: Every factual or numerical statement must include a precise citation (document, section, page, or table reference). If the answer cannot be directly verified in the retrieved text, the model will return “not found in source” instead of fabricating an answer.

Change Tracking Across Filings: Automatically compares sequential filings to detect and highlight newly added or modified risk factors or business narratives. Helps analysts quickly focus on what changed and why.

Lightweight, Modular Design: Although it performs complex reasoning, the system exposes simple APIs that integrate easily with analyst workflows.It supports batch processing of multiple companies and quarters, not just single-query interactions.

Data & Sources

To ensure transparency, reproducibility, and legal compliance, the project will rely entirely on publicly available SEC datasets and open research corpora.

Primary Sources

SEC EDGAR Filings and API: The U.S. SEC provides open access to all company filings (10-K, 10-Q, 8-K, etc.) via the EDGAR search portal and data.sec.gov API. JSON and XBRL endpoints enable structured programmatic access to filing metadata and documents.
SEC Financial Statement Data Sets: Structured quarterly datasets containing normalized financial statement values from 2009 onward. (Freely downloadable at sec.gov/data-research/sec-markets-data/financial-statement-data-sets)
EDGAR-Corpus (Academic Text Dataset): A large-scale academic corpus of annual and quarterly reports pre-segmented by section, used in financial NLP research. (Available at arXiv:2109.14394)
OpenEDGAR Framework: An open-source Python toolkit for downloading, parsing, and managing EDGAR filings. Repository: github.com/datasets/edgar.
Material Contracts Corpus (MCC) *(optional supplemental dataset):* Contains text of corporate contracts disclosed in SEC filings, useful for detecting legal or supply-chain risk shifts. (See arXiv:2504.02864)

Data Use Plan

We will select 10–20 representative public companies and collect their latest 4–8 quarters of 10-K/10-Q filings. Financial statement data will be fetched via SEC's XBRL and Financial Statement Data Set APIs for structured numeric values. Text and tables will be parsed using OpenEDGAR and cleaned into section-tagged chunks for retrieval and model training. All access will comply with SEC rate limits and fair-use guidelines described here.

Technical Approach

Ingestion & Parsing: Download company filings with the ticker, read the HTML or XBRL files, find the main sections, and pull out tables along with their titles and units.
Indexing (RAG): Break each section into smaller parts, add details like section name, time period, and units, and use this info to find more accurate and relevant results.
KPI Normalization: Map label variants (e.g., "Net sales" to "Revenue"), standardize units, and compute common ratios, including margins and free cash flow.
Model Fine-Tuning: Supervised fine-tuning a small open-source model on curated Q&A pairs from filings, optimized for concise and citation-based outputs.
Guardrails: Ensure that every number in the answer is directly derived from a table or quoted text. If the source can't be found, reply with "not found in source."
Deployment: Connect the project to the existing FastAPI backend, package it with Docker, and test its performance both on your computer and in the cloud.

User-Facing Component

Upload or link filings: Users can add a company's financial report to start the analysis.
Ask questions: Users can ask about financial details and get sourced answers.
Generate KPI summaries: The system produces standardized tables of key metrics across periods.
Summarize risks: The tool identifies and categorizes new or changed risk factors.
Interactive web interface: A simple demo page lets users try out all the features in one place.

Evaluation Plan

Grounding & Factuality: Measure Grounded Numeric Rate = (# numeric claims found verbatim in cited span/table) ÷ (total numeric claims). Add targeted spot-checks for non-numeric statements (common errors: mis-scaling, wrong period, mis-attribution).
Retrieval Quality: Create gold Q→span labels and report Recall@k and MRR for the retriever to distinguish indexing/chunking issues from generation errors.

Table Accuracy: After unit/currency normalization, compare extracted KPIs to official tables and report MAPEand MAE, plus an Exact-Match Rate (within ±0.5% tolerance) for core metrics (revenue, COGS, gross profit, operating income, net income, cash flow, debt).
Usability: Analyst task: "Find three drivers of gross margin change." Track time-to-insight, follow-up query count, and confidence (Likert). Collect brief qualitative feedback on citations, table readability, and error messaging.
Ethics & Limits: Document data licensing, risks (hallucinations, stale filings, non-GAAP interpretation), and mitigations (mandatory citations, numeric guardrails, confidence/coverage notes). Clarify non-advisory scope and explicit Non-GAAP labeling.

Team Workflow & Milestones

Weeks 3–5: Retrieval MVP: Ingest filings by ticker/CIK; section-split (Items 1, 1A, 7, 8, Notes) with metadata-rich chunks. Baseline RAG + /ask returning cited answers. Initial eval harness for Recall@k and grounded numeric checks.
Weeks 6–8: Tables & KPIs: Table extraction + normalization (units, currency, label mapping like "net sales"→revenue). /kpi endpoint (CSV/JSON) with provenance. Add MAPE/MAE dashboard and fix top extraction errors.
Weeks 9–11: Fine-Tuning & Risks: SFT a small open LLM on curated filing Q&A for concise, cited responses.Risk extraction + change detection; release /risks. Expand eval: grounding rate on SFT model; retriever ablations (section boosts).
Weeks 12–13: Demo, Benchmark, Write-Up: Dockerized local demo; optional cloud VM latency benchmark.Polish UI/outputs (citation clarity, unit/currency badges, confidence/coverage).Finalize evaluation tables/plots and ethics/limitations; complete report & slides.

Expected Deliverables

GitHub Repo: Modular code (ingest → parse → index → generate), Dockerfile, README with setup/run/eval, and example requests/responses for /ask, /kpi, /risks.
Live Demo (Local Docker): Minimal UI/Swagger showing ingest, grounded Q&A with citations, KPI table extraction (units/currency preserved), and risk summaries with new/changed flags.
Final Report & Slides: Problem framing, data sources, system design, results (Grounded Numeric Rate, Recall@k/MRR, MAPE/MAE, usability), key ablations, ethics/limitations, and prioritized future work (e.g., XBRL-native parsing, multi-company panels, lightweight reward modeling).