Scaling Archival Research with Agentic AI: A Reproducible Pipeline for IPO Team Data Enrichment

Máté Tamás Molnár (s4077741)

Supervisors:
Amirhossein Zohrehvand & Bas Kruiswijk

Bachelor Thesis– Data Science and Artificial Intelligence

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

July 16, 2026

Abstract

Initial Public Offerings (IPOs) trigger significant shifts in organizational structure, making granular Top Management Team (TMT) data crucial for financial research. However, extracting this data from heterogeneous SEC S-1 filings predominantly relies on time-consuming manual extraction. As single-pass Large Language Models (LLMs) often struggle with cognitive overload and context window exhaustion when parsing complex sections in a single call, this thesis introduces a reproducible, two-phase multi-agent Artificial Intelligence (AI) pipeline to automate TMT data extraction from U.S. IPOs. Phase 1 deterministically discovers and classifies 11,251 filings, including withdrawn offerings. Phase 2 benchmarks a 2×2 experimental matrix that crosses extraction methodology (one-shot vs. specialized roster-identification) with self-reflection (no verification vs. reflexive verification), using a year-stratified sample of 150 companies to evaluate accuracy, latency, and token usage. Results show that task decomposition moderately improves extraction quality and substantially increases pipeline robustness, with the specialist extractor producing output for every filing, whereas the one-shot extractor fails on roughly one in five. Furthermore, reflexive verification yields no statistically detectable accuracy gain at roughly triple the token usage. The optimal configuration, specialist extraction without verification, achieves approximately 78% precision and 55% recall on local hardware, offering a scalable, reproducible first pass that reduces manual archival collection without requiring API access.

Contents

1 Introduction

Initial Public Offerings (IPOs) represent the process through which a company transfers its ownership from private to public markets. In the United States, this transition is governed by the Securities and Exchange Commission (SEC), which requires domestic companies to file an S-1 registration statement before offering shares. Since going public triggers documented shifts in leadership composition, innovation strategy, and executive behavior [Chahine and Zhang, 2020, Bernstein, 2012, Chai et al., 2025], granular data on a company's TMT at the time of its IPO is essential for financial and organizational research.

Despite the need for granular TMT data, the reliance on manual data extraction from unstructured SEC filings creates a severe, time-consuming bottleneck. These filings are heterogeneous in structure; the SEC EDGAR database contains over 11,000 S-1 submissions for U.S. operating companies between 2000 and 2020. Prior work has demonstrated that LLMs are creating a viable alternative to the prevalent hand-collection of such granular data [Gilardi et al., 2023]. However, processing full S-1 filings in a single LLM call causes context-window exhaustion and cognitive overload [Kulkarni and Kulkarni, 2026]. Moreover, even after deterministic pre-processing isolates the relevant Management section, prompting a single LLM call to identify every executive, assign their organizational role, and contextualize their biography in one go, still carries a significant cognitive load, motivating the testing of whether decomposing this task across specialized agents improves accuracy.

This paper addresses these bottlenecks through a reproducible, two-phase pipeline. Phase 1 deterministically queries EDGAR, classifies 11,251 S-1 filings by IPO outcome, including withdrawn offerings to mitigate selection bias, and constructs a year-stratified evaluation sample of 150 companies. Phase 2 deploys a 2×2 experimental matrix across two architectural dimensions: extraction methodology (a single generalist LLM call versus a decomposed, specialized roster-identification and labeling pipeline) and system reflexivity (the absence of verification versus the application of a reflexive verifier agent to detect hallucinations and force re-extraction). By isolating these two dimensions, the study quantifies the cost-accuracy trade-off for agentic TMT extraction and determines when the computational overhead of multi-agent orchestration is justified for financial archival research.

2 Theoretical Background

2.1 Strategic and Behavioral Thresholds of the IPO

An IPO has numerous effects that significantly change not only a company's financial position but also aspects such as innovation, executive structures, and even a CEO's social media behavior. Therefore, an IPO is not merely a financial transaction of stakeholder ownership, but a fundamental change in a company's operational structure, underscoring the critical importance of the data extracted in this research.

Innovation and Strategic Realignment. Going public fundamentally shifts a company's approach to innovation. Prior research demonstrates that post-IPO, internal innovation novelty declines and key inventors often leave [Bernstein, 2012]. Consequently, companies pivot from internal

innovation through Research and Development (R&D) to Mergers and Acquisitions (M&As) to acquire external innovation. This strategic pivot requires a corresponding shift in leadership expertise, making it essential to identify Top Management Team (TMT) composition to understand how firms navigate this transition.

The Strategic Option to Withdraw and Mitigating Selection Bias. The transition to public markets is a strategic milestone, not a guaranteed outcome. Companies leverage the threat of withdrawal to negotiate better offer pricing [Busaba et al., 2001]. Therefore, withdrawn IPOs represent strategic exits rather than operational failures. Prior research notes that tracking only successful IPOs introduces selection bias, as changes in leadership or innovation could stem from preexisting firm characteristics rather than the IPO itself. Including withdrawn IPO filings in our dataset directly mitigates this bias [Bernstein, 2012].

Fiduciary Obligations and Executive Behavior. Recent behavioral strategy research highlights the impact of the IPO transition on a company's executives' fiduciary obligations, demonstrating that following an IPO, CEOs shift their social media activity to emphasize company performance while de-emphasizing broader societal issues [Chai et al., 2025]. These findings illustrate that IPO transition is a critical catalyst for behavioral changes at the executive level. Furthermore, it underpins the importance of the granular Top Management Team (TMT) data this agentic extraction pipeline produces, as researchers require exact leadership timelines to study these behavioral shifts.

2.2 Leadership Composition and Organizational Structure

CEO Human Capital and Pre-IPO Gearing. Venture Capitalists (VCs) frequently orchestrate CEO replacements before an IPO to ensure the managerial team can handle the complexities of the public markets [Chahine and Zhang, 2020]. The likelihood of a CEO replacement is highly correlated with the current CEO's human capital and the geographic distance to the lead VC. Because these leadership shifts directly impact IPO valuations and post-IPO performance, extracting granular data on executive education and prior experience is critical for our analysis.

Measuring Structure through TMT Composition. Mapping organization structure is vital for understanding corporate strategy [Albert et al., 2026]. While S-1 filings mandate executive disclosure, the reporting formats are highly heterogeneous. Manual classification is inefficient, necessitating the use of Large Language Models (LLMs) [Gilardi et al., 2023]. This thesis adopts Albert et al.'s taxonomy, which classifies each executive along two dimensions: six managerial role categories with ten functional sub-categories, and twelve hierarchical rank designations. By pairing this taxonomy with agentic extraction, we bridge organizational theory with scalable data parsing.

2.3 Multi-Agent Orchestration and Agentic Task Decomposition

Extracting granular financial data from complex SEC filings exceeds the capabilities of single-agent architectures, which frequently struggle with context-window exhaustion and cognitive overload [Masterman et al., 2024, Sapkota et al., 2026]. To address the schema heterogeneity of regulatory

documents, the AgenticIE framework was developed, which uses a planner-executor-responder loop with shared memory [Colakoglu et al., 2026]. Relying on single-prompt extraction for S-1 statements would introduce severe context overhead. Research has shown that decomposed and least-to-most prompting improves the solving of complex tasks by breaking down complex problems into a sequence of subproblems [Khot et al., 2023, Zhou et al., 2023]. Moreover, the "lost in the middle" phenomenon showed that performance often degrades when models must access relevant information in the middle of longer contexts [Liu et al., 2023]. Thus, to optimize parsing robustness, this research decomposes information extraction into a multi-step pipeline to minimize the context window. Because task decomposition has been shown to improve performance and bounds the search space, specialized architectures, utilizing agentic task decomposition, should outperform single-pass models, leading to the first hypothesis:

> ***Hypothesis 1.*** *Agentic decomposition of the extraction tasks into a specialized pipeline, a roster-identification stage followed by a dedicated role-classification stage, will achieve a higher person-detection accuracy on S-1 filings than a single-pass generalist extraction model that performs identification, classification, and contextualization in one LLM call.*

2.4 Reflexive Verification and Self Correction

Extraction difficulty varies significantly across domains, with the governance and executive compensation sections particularly difficult due to inconsistent formatting [Kulkarni and Kulkarni, 2026]. In financial archival research, data integrity is paramount, making reflexive architectures crucial to stop the unchecked propagation of hallucinations through self-correction. However, research is split on whether self-correction reliably delivers this promise. Iterative self-refinement with self-generated feedback improves output quality across diverse tasks [Madaan et al., 2023]; moreover, research reports a consistent reflexive accuracy premium of approximately +0.04 F1 in structured financial extraction [Kulkarni and Kulkarni, 2026]. Conversely, research demonstrates that LLMs often fail to improve, and even degrade, when correcting their own output without external feedback [Huang et al., 2024]. Importantly, the gains reported by Madaan and Kulkarni were achieved through the utilization of frontier LLMs, capable of reliably identifying errors the extractor could not avoid, whether this transfers to a 7B model is still to be seen.

> ***Hypothesis 2.*** *The removal of the verification (responder) agent from the extraction pipeline will substantially increase the hallucination rate* [1] *and reduce overall field-level extraction accuracy compared to a fully verified architecture.*

2.5 The Economics of Agentic AI: The Cost-Accuracy Trade-Off

While multi-agent systems and reflexive verification loops theoretically improve accuracy, they inherently introduce substantial computational overhead. Orchestrating multiple LLM calls, executing task decomposition, and running iterative self-correction loops dramatically increase both token usage and end-to-end latency (wall-clock execution time) compared to standard one-shot models. However, because financial archival research requires audit-level accuracy to prevent the introduction of systematic bias into the datasets, the marginal gain in extraction trustworthiness is expected to outweigh the computational cost, leading to the final hypothesis:

[1] See Results section for definition.

Hypothesis 3. *The full agentic pipeline (specialist extraction with reflexive verification) will achieve a statistically meaningful improvement in person-detection F1 over the one-shot baseline, at a token cost premium* [2] *that remains proportionate to the accuracy gain.*

3 Methodology and Data Architecture

3.1 Research Design and Empirical Framework

To test the hypotheses developed above, this research presents a reproducible framework to isolate the mechanisms that make Large Language Model (LLM) extraction reliable in financial domains. To achieve this, this research adapts the AgenticIE multi-agent framework to the financial landscape, constructing a robust two-phase data architecture designed to decouple data retrieval from experimental extraction:

- **Phase 1** functions as a deterministic discovery and labeling pipeline.
- **Phase 2** deploys a 2×2 experimental matrix to evaluate distinct extraction and verification configurations against a hand-verified ground truth dataset.

Ultimately, this research evaluates the resulting pipeline across three critical dimensions: extraction accuracy, wall-clock latency, and token usage. By tracking these metrics, this study quantifies the cost-accuracy trade-off and determines whether the accuracy gains from multi-agent verification are proportional to the computational overhead in financial archival research.

All experiments run locally through the open-weight model `Qwen2.5:7b` (Qwen2 architecture, 7.6B parameters, `Q4_K_M` 4-bit quantization, Apache-2.0), served locally through Ollama 0.30.11, with a context window of 8,192 tokens. This model was selected so that the entire setup is openly available and version-pinned for independent re-execution without API access, and so confidential pre-IPO disclosures never leave local hardware.

3.2 Data Collection and Sample Selection

The initial sample comprises S-1 registration statements for U.S. Initial Public Offerings filed between 2000 and 2020. The discovery pipeline utilizes the SEC EDGAR database via the `edgartools` Python library to query all relevant filings within this temporal window. Following established methodology [Bernstein, 2012], entities with SIC codes between 6000 and 6999 are excluded; the remaining Bernstein exclusion criteria (closed-end funds, ADRs, limited partnerships) are satisfied implicitly by only querying S-1 registration statements, which these entity types do not use. Special Purpose Acquisition Companies (SPACs), which do file S-1s, were not explicitly filtered. Furthermore, duplicate filings resulting from pre-IPO amendments (S-1/A) are consolidated, retaining only the accession number of the most recent filing prior to the outcome date. It is important to note that this discovery pipeline represents a temporal snapshot executed on 2026-06-24, at 18:09

[2] Cost is determined as the ratio of mean token usage (input + output tokens) between the full and none verification modalities within the same extraction arm. Verification is justified by cost, if and only if it produces a statistically detectable F1 gain.

CEST; consequently, outcome classification reflects the filing status as of this date.

To integrate established academic databases, the pipeline extracts the 9-digit CUSIP identifier for each remaining entity. Since S-1 statements do not consistently list CUSIPs, this extraction targets the entity's corresponding SC 13G filings (institutional ownership disclosures), where the identifier appears in a structured format. The pipeline downloads the first 10KB of the most recent SC 13G filing and applies two sequential regex patterns: an SGML-tag pattern and a cover-page free-text pattern. If both fail, a local LLM call is used as a fallback. Entities for which no SC 13G filing exists receive a null CUSIP and cannot be matched via the Ritter database; their classification falls back to EDGAR filing status.

As a result of the filtering logic, the final state machine (`manifest.db`) yielded a dataset of 11,251 unique S-1 registration statements ready for agentic extraction.

3.3 Variable Operationalization and Extraction Schema

To translate unstructured S-1 management sections into a quantifiable dataset, this pipeline incorporates a standardized executive classification taxonomy [Albert et al., 2026]. TMT characteristics are represented by distinct variables, encompassing three primary domains: human capital attributes (e.g., age, prior education, prior experience), managerial role and hierarchical rank (e.g., CEO role flag, SVP rank designation, board membership), and functional role sub-classifications (e.g., Primary Value Chain, Support Functions).

Rather than relying on the agent's semantic interpretation, to guarantee reproducibility, the extraction is constrained using strict rules defined by Pydantic models. The schema uses three classes for the domains highlighted above: the `CompanyInfo` class captures company-specific metadata (e.g., CUSIP, CIK, SIC, outcome status) and contains a list of `Executive` objects. Each `Executive` object enforces boolean flags for roles and integrates `PrimarySubcategory` and `SupportSubcategory` classes intended to capture mutually exclusive and collectively exhaustive functional roles.

Furthermore, the pipeline explicitly quantifies its uncertainty by the Verifier agent evaluating the Extractor agent's output against the source text. It verbalizes its confidence through a confidence score. The confidence threshold of 0.85 is adopted from Kulkarni and Kulkarni, without recalibration. When the confidence score falls below this threshold, the Verifier withholds the data, generates a critique, and forces a re-extraction attempt, with the Verifier agent's itemized critique appended as guidance to the next extraction call. If extraction fails to meet the confidence threshold after the maximum retry budget of 3 is exhausted, the pipeline returns the unverified, best-effort extraction, and only the specifically ungrounded fields are returned as missing (`NaN`).

4 The Two-Phase Information Extraction Pipeline Implementation

4.1 Phase 1: Deterministic IPO discovery and Output Labeling

Even though the nature of the required data is known beforehand, automatic, programmatic access to these documents poses significant bottlenecks that require robust web scraping. To query the SEC's database without violating its strict 10 requests per second rate limits [U.S. Securities and Exchange Commission, 2026], the pipeline waits 0.2 seconds between company-level requests and 0.5 seconds between quarterly batches, ensuring uninterrupted data retrieval.

The deterministic data plumbing pipeline queries the SEC EDGAR database for all S-1 filings in the given date range. During extraction, based on selection and matching criteria highlighted earlier in section 3.2, we filter out unwanted entities and cross-reference successful ones. Once the filtering and cross-referencing are complete, the pipeline moves on to classifying the IPOs based on success in the following manner:

- **Successful IPOs:** Companies present in the Ritter database are classified as successful. As a fall-back for IPOs not yet indexed in the Ritter database, entities are queried against EDGAR; those having a `EFFECT` (Notice of Effectiveness) filing (filed by the SEC) are also classified as successful.
- **Withdrawn IPOs:** Entities not present in the Ritter database, but have an `RW` (Registration Withdrawal) filing, are classified as withdrawn. The inclusion of such unsuccessful offerings is a fundamental methodological design choice to mitigate the selection bias prevalent in earlier IPO research [Bernstein, 2012].
- **Stalled IPOs:** Entities without both `EFFECT` and `RW` filings, and not present in the Ritter database, are classified as stalled, representing active but unpriced registrations at the time of the discovery date.

The classification yielded 6734 successful IPOs, 3192 withdrawn offerings, and 1325 stalled registrations out of the 11,251 total number of IPOs identified in Phase 1.

The pipeline is governed by an SQLite-based state machine (`manifest.db`) that serves as the main orchestrator for the dataset. In contrast to holding data in memory, the manifest continuously logs company metadata (e.g., CIK, parsed filing URLs, etc.), output classifications, and the execution state for Phase 2. Each document is assigned a processing status flag (`PENDING`, `DONE`, `FAILED`, or `HELD`), allowing the pipeline to resume mid-run without reprocessing completed entries.

4.2 Phase 2: Agentic Pipeline and Experimental Extraction Design

To evaluate the impact of the architectural modifications on dataset quality and computational efficiency, Phase 2 implements a multi-dimensional evaluation baseline. The primary architectural question of this thesis is whether adding a reflexive verification loop to an otherwise linear extraction

pipeline will significantly improve data integrity. Furthermore, testing task decomposition (One-Shot vs. Specialist) alongside this loop determines how cognitive bounding interacts with self-correction.

4.2.1 The Core Agentic Components

To address the schema heterogeneity of regulatory documents, the AgenticIE framework was developed using a planner-executor-responder loop with shared memory [Colakoglu et al., 2026], decoupling the extraction task into the following three specialized nodes:

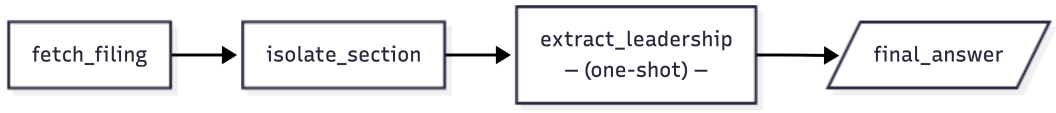
- **The Planner:** The Planner coordinates tool execution according to a fixed sequence policy: it first fetches the target SEC filing, then deterministically isolates the management section (reducing the context window), and finally dispatches the configured extraction tool. A loop-detection guard prevents the Planner from reissuing identical tool calls.
- **The Executor:** Receiving the isolated sections from the Planner, the Executor executes targeted LLM calls to extract relevant data, while formatting it along the pre-defined `Pydantic` schemas defined in 3.3.
- **The Responder (Verifier):** Cross-referencing the structured output produced by the Executor against the original raw text, the Responder acts as a quality gate. Hallucinations are flagged, uncertainty is quantified via provided confidence scores, and `JSON` outputs are either finalized or returned to the Planner with a critique to re-initiate the extraction loop.

4.2.2 The 2×2 Experimental Matrix

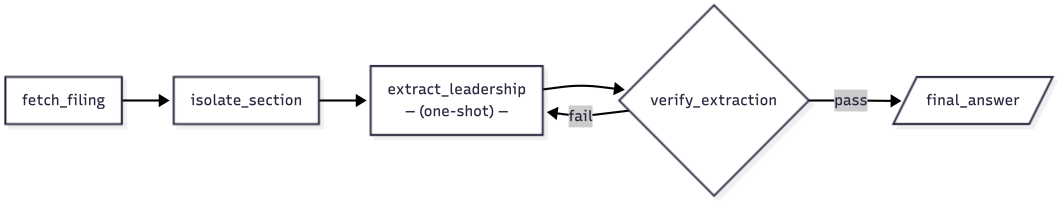
To isolate the performance gains generated by multi-agent orchestration from the raw semantic capabilities of the underlying language model, the pipeline evaluates four distinct orchestration topologies:

1. **The Baseline (One-Shot + No Verification):** This one-shot extraction model establishes the non-agentic baseline. The LLM is instantiated with a singular system prompt that tasks it with parsing the relevant S-1 section and extracting all TMT variables in a single, linear execution step. It lacks both task decomposition and self-reflection.
2. **The Extraction Ablation (Specialist + No Verification):** This configuration tests the isolated effect of task decomposition. The extraction is decoupled into two sequential LLM tool invocations: a Roster stage that identifies names, titles, and biographical context, followed by a Label stage that assigns the role and rank flags from the Albert et al. taxonomy. No verification pass is applied, so extraction errors are not caught or corrected.
3. **The Architectural Modification (One-Shot + Full Verification):** This tests whether adding a reflexive self-correction loop can salvage the cognitive overload of a one-shot prompt. The LLM extracts the data in a single pass. However, a Responder (Verifier) agent cross-references the output against the raw text, returning critiques and forcing retries if hallucinations are flagged. Unlike the specialist architecture, this verification loop is implemented directly into the sequential one-shot path rather than relying on the agentic Planner for routing.

4. **The Full Agentic Pipeline (Specialist + Full Verification):** This represents the complete proposed architecture. It combines the cognitive bounding of the Planner and Executor with the strict quality control of the Responder, initiating an iterative loop until the data meets the required confidence threshold, or the system exhausts available attempts.

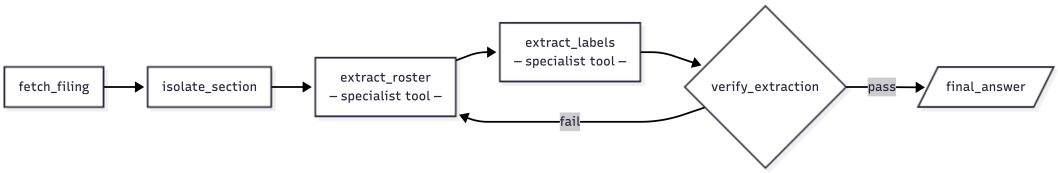


(a) Generalist, No Verification



(b) Generalist + Full Verification

(c) Specialist, No Verification



(d) Specialist + Full Verification

Figure 1: Tool-level execution flows for each cell of the experimental matrix.

Each extraction arm is operationalized through a distinct zero-shot prompt design. The generalist system prompt instructs the model to identify, classify, and contextualize every executive in a single call, providing the full Albert et al. taxonomy inline as classification rules. The specialist arm decouples extraction into two focused prompts. First, the roster prompt instructs the model to identify names, titles, and biographical detail only, while prohibiting any classification. Second, the label prompt then receives the roster and applies the taxonomy to each entry by index. The verifier prompt instructs the model to flag only factual hallucinations and omissions (e.g., invented names, wrong ages, or missing officers) while explicitly ignoring title-wording differences, and also to return a confidence score alongside errors.

To avoid context fragmentation, the architecture deploys a global blackboard pattern. The two main objects are: `AgentState`, which caches intermediate results, raw document contexts, and telemetry (retry counts, wall-clock time, cumulative token usage); and `AgentStatus`, a five-valued control flag (`PLAN`, `NEED_TOOL`, `RESPOND`, `SUCCESS`, `END`) embedded within `AgentState` that drives

state-machine transitions between pipeline stages. By continuously updating the manifest database with these logs, the system ensures traceability for every extracted TMT variable.

5 Analytic Strategy and Evaluation Methodology

To validate the performance of the implemented architectures, the system's outputs are benchmarked against a ground truth dataset. This dataset encompasses 2,157 U.S. IPOs filed between 2000-07-31 and 2020-11-25, serving as the baseline for extraction accuracy.

5.1 Sampling Strategy and Methodological Justification

To effectively evaluate the extraction pipeline without compromising statistical validity, this research utilizes a hand-collected ground truth dataset of 2,157 scored U.S. IPOs (filed 2000–2020) provided by the project supervisor; even though scoring methodology is undocumented, this dataset is treated as the gold standard for benchmarking. A year-stratified sample of $N = 150$ companies was constructed from the intersection of the ground truth dataset and the pool of discovered S-1 filings. Stratifying the sample by year directly mitigates discovery bias. Because the availability and formatting of EDGAR filings vary over time, a simple random sample from the overlapping pool would be inherently skewed toward eras with higher discoverability.

The sample size of $N = 150$ was chosen to balance the computational cost (each document is processed under four pipeline configurations, yielding 600 total extraction runs) against sufficient coverage across the yearly strata. The resulting year distribution is reported in Table 1. Allocation is proportional to each year's share, so companies carry equal weight in the evaluation.

5.2 Reproducibility and Cost Tracking

For every execution, the evaluation pipeline aggregates cumulative token usage and wall-clock execution times to quantify the architecture's feasibility. Configured for reproducibility, all experiments utilize an LLM temperature of 0.0, enforce a fixed random seed (42), and record exact execution timestamps and intermediate extraction states within the `manifest.db` tracker.

6 Results

The results are evaluated across two scopes. The *common-subset* scope ($n = 110$) isolates extraction *quality* by restricting scoring to companies for which all modalities produced output. The *end-to-end* scope ($n = 137$) measures the performance of the pipeline as deployed, through counting a modality's failure to produce output as 0 recall for that company. Macro-averaged metrics are reported as the primary result because the company is the unit of sampling and analysis; micro-averaged metrics are reported for contrast. Uncertainty is quantified using 95% company-clustered bootstrap confidence intervals, consistent with the bootstrap reporting in [Kulkarni and Kulkarni, 2026] (1,000 resamples, seed 42); paired per-company ΔF1 with bootstrap CIs are reported for each contrast of interest. Hallucination rate is inferred from false positives, extracted executives with no ground-truth counterpart during extraction, i.e. hallucination rate = 1−person precision.

Year	GT pop.	GT %	Pool	Sample	Sample %
2000	1	0.0	0	0	0.0
2004	1	0.0	1	0	0.0
2005	42	1.9	32	3	2.0
2006	183	8.5	153	13	8.7
2007	192	8.9	148	13	8.7
2008	75	3.5	53	5	3.3
2009	81	3.8	56	6	4.0
2010	199	9.2	138	14	9.3
2011	197	9.1	140	14	9.3
2012	127	5.9	82	9	6.0
2013	175	8.1	141	12	8.0
2014	199	9.2	157	14	9.3
2015	125	5.8	110	9	6.0
2016	89	4.1	82	6	4.0
2017	103	4.8	94	7	4.7
2018	142	6.6	129	10	6.7
2019	103	4.8	93	7	4.7
2020	123	5.7	110	8	5.3
Total	2,157	100.0	1719	150	100.0

Table 1: Year-stratified sampling distribution of S-1 filings (2000–2020)

6.1 Extraction Quality on the Common Subset

Table 2 summarizes the person-detection F1 across the four cells. Both specialist cells (0.621, 0.614) exceeded both generalist cells (0.590, 0.610). The two verification deltas carry opposite signs, verification improves the generalist by +0.020 and worsens the specialist by −0.007.

Table 3 decomposes the F1 scores into precision and recall, revealing that precision remains effectively constant across all configurations (0.763–0.793), meaning that the variation in overall F1 performance is driven by changes in recall.

Table 4 reports macro-averaged metrics with 95% company-clustered bootstrap confidence intervals. Macro and micro estimates are close respectively (e.g., 0.609 vs. 0.614 for the specialist with full verification), indicating that the findings are not driven by filings containing more executives.

6.2 Hypothesis Tests: Paired Comparisons

Extraction Effect (H1). Without verification, the specialist outperforms the generalist by a mean per-company ΔF1 of +0.026 (95% CI $[+0.003, +0.051]$), scoring higher on 18, lower on 11 and tying on 81 companies. The CI barely excludes zero, indicating a modest effect; therefore, Hypothesis 1 is supported in the configuration without verification. Under full verification the effect shrinks to +0.006 (95% CI $[-0.026, +0.038]$), the interval spans zero and Hypothesis 1 cannot be confirmed in that arm. Notably, field-level extraction quality follows this same trend; Position F1 moves in the same direction (moving from 0.745 for the generalist baseline to 0.777 for the specialist without verification) but it was not formally statistically tested.

Verification effect (H2). Verification did not produce detectable accuracy change, with +0.014 (95% CI $[-0.008, +0.039]$) for the generalist and −0.006 (95% CI $[-0.026, +0.015]$) for the specialist. While the verifier was invoked for every successfully extracted company and re-extraction was triggered for 64% of verification events in the generalist arm and 76% in the specialist arm, its corrections did not consistently improve alignment with the ground truth. Thus Hypothesis 2, predicting a substantial accuracy loss from removing verification, is not supported.

6.3 End-to-end Pipeline Performance

The common-subset analysis is based on successful completion, obscuring an important architectural difference. The generalist configurations failed to produce output for 27 of 137 companies (19.7%), while the specialist configurations completed all 137. These failures mainly arose due to context window limit exhaustion, stemming from the prompt length of generalist extraction, whereas decomposition also allowed for shorter system prompts, remaining within budget on the same filings.

Table 6 reports end-to-end macro metrics with failures scored as zero recall. While generalist precision is unaffected, since failures do not produce false positives, generalist recall drops from $\sim$0.51 to $\sim$0.41 and F1 to 0.473–0.484. Conversely, the specialist recall minimally increases from $\sim$0.54 to $\sim$0.55 and F1 to 0.618–0.632. Table 7 reports the paired end-to-end comparisons, showing that extraction effect grows to ΔF1 = +0.159 (95% CI $[+0.112, +0.215]$) without verification,

and +0.134 (95% CI [+0.085, +0.188]) with verification; strongly supporting Hypothesis 1 on the end-to-end scope. However, verification CIs again span zero in both arms.

6.4 Computational Cost and the Cost-Accuracy Trade-Off (H3)

Table 8 reports average token usage and wall-clock time per company with the corresponding accuracy in both scopes. Within each extraction modality, verification multiplied token usage by 3.8× (generalist) and 2.7× (specialist), while increasing latency by 2.7× (generalist) and 2.3× (specialist), with no accuracy improvement within bootstrap CI margins.

When evaluated across the end-to-end scope, the specialist pipeline demonstrates significantly higher effectiveness than the generalist baseline. While the common-subset performance deltas remain marginal, the end-to-end evaluation reveals an order-of-magnitude improvement in robustness for the specialist configuration, mitigating the high failure rates observed in the generalist arm.

Crucially, the full pipeline (specialist + verification), at 7.8 × baseline cost and 4.6× latency increase, yields no accuracy improvement over the specialist-only configuration, with CIs for both differences spanning zero. Hypothesis 3 is therefore rejected; while the specialist-none configuration provides a statistically detectable accuracy gain over the baseline, the reflexive verification component provides no such improvement, rendering the total pipeline cost disproportionate to its performance. The rational configuration choice becomes the specialist without verification, operating at roughly 13,000 tokens and 53 seconds per filing on local hardware. Ultimately, as shown in Table 7, the specialist with verification significantly outperforms the baseline by ΔF1 = +0.145 (95% CI [+0.097, +0.197]), confirming that while the verification component itself buys no accuracy, the agentic pipeline as a whole provides substantial robustness gains over the one-shot baseline.

	None	**Full**	Δ (Full−None)
Generalist	0.590	0.610	+0.020
Specialist	0.621	0.614	−0.007
Δ (Spec−Gen)	+0.031	+0.004	

Table 2: Performance metrics for the common-subset evaluation (n=110). Values represent micro-averaged F1 scores pooled across all extracted executives.

Variant	**P**	**R**	**F1**	**Pos F1**
Generalist + None	0.763	0.482	0.590	0.745
Generalist + Full	0.793	0.496	0.610	0.743
Specialist + None	0.764	0.524	0.621	0.777
Specialist + Full	0.771	0.510	0.614	0.752

Table 3: Micro-averaged extraction metrics on the common subset (110 companies). P = Person Precision, R = Person Recall, F1 = Person F1, Pos = Position F1.

Variant	Precision	Recall	F1
Generalist + None	0.774 [0.715, 0.832]	0.508 [0.457, 0.559]	0.589 [0.539, 0.640]
Generalist + Full	0.796 [0.741, 0.849]	0.519 [0.469, 0.568]	0.603 [0.554, 0.655]
Specialist + None	0.762 [0.700, 0.820]	0.543 [0.494, 0.590]	0.615 [0.562, 0.665]
Specialist + Full	0.776 [0.718, 0.829]	0.534 [0.481, 0.583]	0.609 [0.560, 0.657]

Table 4: Macro-averaged person-detection precision, recall, and F1 with 95% company-clustered bootstrap confidence intervals (1,000 resamples, common subset, $n = 110$).

A	B	ΔF1	95% CI	W/L/T
Verification effect (within extraction arm)				
Gen Full	Gen None	+0.014	[−0.008, +0.039]	13/5/92
Spec Full	Spec None	−0.006	[−0.026, +0.015]	11/11/88
Extraction effect (within verification arm)				
Spec None	Gen None	+0.026*	[+0.003, +0.051]	18/11/81
Spec Full	Gen Full	+0.006	[−0.026, +0.038]	16/20/74
Cross-arm comparisons				
Spec None	Gen Full	+0.011	[−0.015, +0.039]	17/20/73
Spec Full	Gen None	+0.020	[−0.009, +0.053]	18/17/75

Table 5: Paired per-company ΔF1 comparisons on the common subset (n=110). CIs are 95% paired bootstrap intervals (1,000 resamples, seed 42). W/L/T = companies where A scored higher / lower / tied vs B. * CI excludes zero.

Variant	Precision	Recall	F1	Coverage
Generalist + None	0.774 [0.718, 0.831]	0.408 [0.357, 0.460]	0.473 [0.417, 0.531]	110/137
Generalist + Full	0.796 [0.741, 0.849]	0.417 [0.365, 0.472]	0.484 [0.429, 0.541]	110/137
Specialist + None	0.784 [0.732, 0.832]	0.555 [0.513, 0.599]	0.632 [0.588, 0.674]	137/137
Specialist + Full	0.789 [0.739, 0.840]	0.545 [0.502, 0.587]	0.618 [0.575, 0.660]	137/137

Table 6: End-to-end macro metrics with 95% company-clustered bootstrap CIs (n=137, 1,000 resamples, seed 42; generalist failures counted as recall = 0).

A	B	ΔF1	95% CI	W/L/T
Verification effect				
Gen Full	Gen None	+0.011	[−0.007, +0.031]	13/5/119
Spec Full	Spec None	−0.013	[−0.037, +0.006]	14/15/108
Extraction effect				
Spec None	Gen None	+0.159*	[+0.112, +0.215]	45/11/81
Spec Full	Gen Full	+0.134*	[+0.085, +0.188]	43/20/74
Cross-arm baseline comparison				
Spec Full	Gen None	+0.145*	[+0.097, +0.197]	45/17/75

Table 7: Paired per-company ΔF1 comparisons end-to-end (n=137, 1,000 resamples, seed 42). Generalist failures are treated as $F1$=0. * CI excludes zero. Results demonstrate that the specialist with verification significantly outperforms the baseline by ΔF1 = +0.145 (95% CI [+0.097, +0.197]).

Variant	Tokens/co.	Rel. cost	s/co.	Rel. latency	F1 (common)	F1 (e2e)
Generalist + None	4.4k	1.0×	26.3	1.0×	0.589	0.473
Generalist + Full	16.5k	3.8×	71.6	2.7×	0.603	0.484
Specialist + None	12.9k	2.9×	52.8	2.0×	0.615	0.632
Specialist + Full	34.4k	7.8×	121.6	4.6×	0.609	0.618

Table 8: Computational cost and end-to-end accuracy per configuration. Tokens are mean per company (input + output) across all attempted companies; relative cost is the token ratio to the cheapest cell.

Variant	Position F1	Doc (strict)	Doc (relaxed)
Generalist + None	0.745	0.009	0.064
Generalist + Full	0.743	0.009	0.064
Specialist + None	0.777	0.009	0.064
Specialist + Full	0.752	0.009	0.073

Table 9: Field- and document-level accuracy (common subset, n=110). Strict document accuracy requires every field of every executive to be correct; relaxed accuracy requires only the executive's presence to be correct.

Verifier Confidence Score	< 0.85 (fail)	$= 0.85$	> 0.85
Generalist + Full	140 (64%)	49 (22%)	30 (14%)
Specialist + Full	226 (76%)	49 (16%)	23 (8%)

Table 10: Distribution of verifier confidence scores across all verification events (219 gen, 298 spec). The modal outcome for both extraction modalities is below the 0.85 acceptance threshold, indicating frequent re-extraction triggers. A notable cluster of observations remains exactly at 0.85 (22% gen, 16% spec), consistent with observations of miscalibrated verbalized confidence [Xiong et al., 2024].

7 Discussion

7.1 Main Findings

The experimental matrix addresses the three hypotheses. Firstly, hypothesis 1 is supported: agentic task decomposition modestly improves mean F1 scores for the common subset (+0.026) and substantially for the end-to-end scope (+0.159). Interestingly, the crucial improvement is not in extraction quality but in robustness (the specialist arm produces output for 27 more companies than the generalist arm). This improvement is mainly attributable to the generalist's context window exhaustion. The generalist's single prompt must carry both the management section and the full Albert et al. schema specification. Due to local LLM and GPU constraints, the context window was set to 8k tokens. In many cases, this was too small for the prompt, and the extraction failed.

Secondly, Hypothesis 2 is not supported: there is no detectable accuracy gain from verification, even though re-extraction was fired for a majority of verification events (64% in the generalist arm and 76% in the specialist arm). While the verification correctly identifies omissions, re-extraction tries to overachieve by extracting every name it finds, without distinguishing between personal names and names in general. Moreover, in the generalist arm, 22% and in the specialist arm, 16% of confidence scores were exactly at the acceptance threshold (Table 10), while not the modal outcome, this clustering at the threshold still remains indicative of miscalibrated verbalized confidence.

Finally, hypothesis 3 is also rejected. While the full pipeline yields a statistically significant accuracy gain over the baseline (ΔF1 = +0.145 (95% CI [+0.097, +0.197])), the verification component itself fails to provide a measurable improvement in accuracy. Consequently, the additional computational overhead required to execute the full verification loop results in a disproportionate cost-to-performance ratio, despite the total pipeline's robust performance.

Notably, the verification deltas carry opposite signs across extraction arms, a pattern consistent with the specialist's decomposition leaving less residual error for the verifier to correct, examined further in Section 7.2.

7.2 The Failure of Verification

As mentioned previously, the verification arm of the pipeline failed, which can be attributed to three mechanisms. Firstly, re-extraction traded precision for recall by recovering some missed executives while over-extracting non-executive names, producing false positives that outweighed the resolved false negatives. Secondly, the verifier's confidence signal proved to carry weak information. Research has shown that verbalized confidence tends to be poorly calibrated in LLMs [Kadavath et al., 2022, Xiong et al., 2024], consequently, 22% of generalist and 16% of specialist confidence scores across all verification events were exactly the 0.85 acceptance threshold (Table 10). Although the model's primary tendency is to trigger re-extraction, this suggests that, when the model is uncertain, it defaults to the minimum required confidence, behaving as a partial binary trigger. Thirdly, as research has shown, intrinsic self-correction often fails without reliable external feedback [Huang et al., 2024]. Moreover, across frontier and 70B-class open-weight models, Kulkarni and Kulkarni reported a +0.04 increase, whereas this research utilizes 7B-parameter local models [Kulkarni and Kulkarni, 2026], which explains the verification null phenomenon.

A secondary pattern emerging is that verification effects differ by extraction arm: the generalist

carries a small positive delta (+0.014) while the specialist carries a small negative one (-0.006) (Table 5). This is likely due to the specialist task decomposition, by splitting identification and labeling into two calls, the specialist reduces the residual errors the verifier could correct. Therefore, the verifier finds fewer omissions and is more likely to trigger unnecessary re-extraction, producing the false positives described earlier. By the generalist making more errors in a single pass, it provides the verifier more signals to act on. This interaction was not anticipated in the experimental design and should be treated as exploratory.

7.3 Decomposition for Robustness, Not Quality

Across both scopes, precision was effectively constant, indicating that architectural changes do not affect the trustworthiness of extracted executives. Variance in F1 came from recall. More importantly, the generalist failed on 27/137 filings, while the specialist produced output for all 137 evaluable filings [3]. As mentioned earlier, this gap is context-window-dependent, and we expect that with a larger context-window budget, it narrows and might even disappear. While the common-subset scope measured output quality, the end-to-end scope measures deployment yield, which is arguably more important for archival researchers than the former.

7.4 Practical Implications

For financial archival researchers, the findings translate into a concrete deployment recommendation. The specialist extraction, without verification, using roughly 13,000 tokens and 53 seconds per filing on local hardware, achieves approximately 78% precision and 55% recall. This makes it suitable for high-precision first-pass processing, eliminating the bulk of manual reading while flagging where human review remains necessary. Moreover, the missed ∼45% of executives likely mainly correspond to document regions the pipeline never reads, so the researcher knows where to look rather than having to re-read the entire filing.

Because the pipeline runs on a locally hosted open-weight model without API access, it is reproducible without subscription cost or data-sovereignty concerns, an important consideration when processing sensitive pre-IPO disclosures. The full 11,251-filing manifest produced in Phase 1, covering both successful and withdrawn offerings from 2000–2020, is available as a validated starting point for researchers wishing to scale extraction beyond the 150-company evaluation sample.

7.5 Limitations

Model scale and local constraints. All results derive from a single 7B open-weight model (Qwen2.5:7B) running within an 8,192-token context window, the largest setting with reliable headroom on shared 12 GB GPUs under multi-tenant load. Both limitations interact: the verification null should not be read as a verdict on reflexive architectures generally, but as a small-model result, Kulkarni and Kulkarni report a consistent +0.04 F1 gain at frontier and 70B-class scale, and without additional points on the scale axis the capability boundary at which verification begins to pay cannot be identified. Similarly, the 27 generalist context failures, and by extension the

[3] 11/150 companies were not evaluable due to missing ground-truth entries; another 2 companies failed in every configuration and were thus removed from the universe.

magnitude of the end-to-end gap, are contingent on the 8k budget; a larger window would reduce or eliminate them.

Recall ceiling. Across scopes, recall is capped around 0.50–0.55, without any architecture moving it significantly. This recall ceiling exists because the ground-truth dataset includes executives named outside the isolated management section. The section-bound extractor cannot recover data from other sections; therefore, this ceiling binds every configuration in the matrix. This upper bound can be improved in future work by implementing document-spanning routing, rather than by improving the current extraction implementation.

Field-level gaps. Age F1 is effectively 0 across the whole matrix, which seems to be a programmatic issue rather than an architectural one. This issue was not resolved within the project timeline. However, it is attributed to two possibilities: first, age is often represented in tables, whose structure might be lost during section isolation; second, the age attribute is under-specified compared to other fields. Regardless, the age dimension of the Albert et al. schema is unmeasured in this study; thus, any claim about the pipeline's completeness across the full document should be read with this caveat in mind.

Small-scale agency. The 7B model required deterministic routing; it failed to follow a four-step plan on its own and did not fire verification. Therefore, the system exercises agency in extraction and self-correction, but not in runtime planning.

Reproducibility. Temperature=0.0 and seed=42 were set for reproducibility; however, local inference with Ollama is not perfectly deterministic across runs. Minor variation exists between runs on identical hardware, but using a fixed temperature and seed minimizes this effect.

7.6 Future Work

Document-spanning routing. To further utilize the power of agentic AI, the pipeline should be reconfigured to implement targeted extractions. This integration could also lift the apparent recall ceiling and provide a configuration that allows to completely replace manual extraction and supervision.

Model-scale replication. To uncover the true model scale where reflexive self-correction starts to pay, the pipeline and evaluation should be re-run using a set of models between the 7B and 70B scale.

Context-window replication. To measure the effect of context window exhaustion on extraction quality, the application should be re-run using various context window sizes (e.g., 16/32k). Moreover, to uncover the precise interaction of context window truncation, which part of the isolated section can be truncated for best performance on filings exceeding the context windows (i.e., first third, second third, or last third), the pipeline should be rerun with varying context window truncation configurations.

External verification. To improve the verifier's performance, consistently with recent research, the verifier agent should be rewired to use a different, possibly cheaper, model; mitigating over-confidence of intrinsic self-correction, providing better feedback for the pipeline.

8 Conclusion

This research aimed to determine whether agentic AI can make the extraction of Top Management Team data from SEC S-1 filings reliable and affordable enough to improve the scalability and timeliness of financial archival research. In the spirit of open science, the resulting pipeline targets data that otherwise exists only as costly, hand-collected proprietary datasets. It is built exclusively on a locally hosted open-weight model, ensuring the entire process is reproducible without API access.

The contribution is a reproducible two-phase pipeline. Phase 1 is a deterministic discovery process that classified 11,251 S-1 registrations filed between 2000 and 2020 by IPO outcome.[4] Phase 2 is an agentic extraction process implementing the Albert et al. labeling schema, evaluated as a 2×2 matrix over two agentic design dimensions: task decomposition and reflexive verification, measuring against one-shot extraction. Performance was measured on a year-stratified sample of 150 companies against a hand-collected ground-truth dataset, using company-clustered bootstrap inference.

The results show that agentic task decomposition earns its overhead: the specialist configuration extracts more executives than the one-shot baseline on the same filings. It produces output for every evaluable filing, whereas the one-shot extractor failed on 27 companies under the same context budget. Even though the self-correction loop fired, flagged omissions, and triggered re-extraction for a substantial majority of verification events (64% for the generalist and 76% for the specialist), reflexive self-correction produced no detectable accuracy change within CI margins at roughly triple the token usage. Autonomous planning proved infeasible at the 7B scale and was replaced by deterministic routing.

As detailed in section 7.4, the optimal configuration is a viable first pass for archival researchers today. Implementing document-spanning routing, lifting the ∼55% recall ceiling, and locating at what model scale reflexive verification begins to pay are the natural next steps, before scaling the validated configuration from the 150-company evaluation sample to the full 11,251 filings.

References

[Albert et al., 2026] Albert, D., Eklund, J. C., and Tang, L. (2026). A new organizational structure database: Examining structure through top management team compositions. *Strategic Management Journal*, 47(3):860–892.

[Bernstein, 2012] Bernstein, S. (2012). Does going public affect innovation? *SSRN Electronic Journal*.

[4]Importantly, Phase 1 classifies 3,192 withdrawn filings, enabling future research on this data to mitigate the selection bias prevalent in prior IPO samples.

[Busaba et al., 2001] Busaba, W., Benveniste, L., and Guo, R.-J. (2001). The option to withdraw ipos during the premarket: Empirical analysis. *Journal of Financial Economics*, 60:73–102.

[Chahine and Zhang, 2020] Chahine, S. and Zhang, Y. A. (2020). Change gears before speeding up: The roles of chief executive officer human capital and venture capitalist monitoring in chief executive officer change before initial public offering. *Strategic Management Journal*, 41(9):1653–1681.

[Chai et al., 2025] Chai, S., Doshi, A. R., and Zohrehvand, A. (2025). Ceo responsibilities and social media activity. Working paper.

[Colakoglu et al., 2026] Colakoglu, G., Solmaz, G., and Fürst, J. (2026). Agenticie: An adaptive agent for information extraction from complex regulatory documents.

[Gilardi et al., 2023] Gilardi, F., Alizadeh, M., and Kubli, M. (2023). Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

[Huang et al., 2024] Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., and Zhou, D. (2024). Large language models cannot self-correct reasoning yet.

[Kadavath et al., 2022] Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., and Kaplan, J. (2022). Language models (mostly) know what they know.

[Khot et al., 2023] Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., and Sabharwal, A. (2023). Decomposed prompting: A modular approach for solving complex tasks.

[Kulkarni and Kulkarni, 2026] Kulkarni, S. and Kulkarni, Y. (2026). Benchmarking multi-agent llm architectures for financial document processing: A comparative study of orchestration patterns, cost-accuracy tradeoffs and production scaling strategies.

[Liu et al., 2023] Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2023). Lost in the middle: How language models use long contexts.

[Madaan et al., 2023] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., and Clark, P. (2023). Self-refine: Iterative refinement with self-feedback.

[Masterman et al., 2024] Masterman, T., Besen, S., Sawtell, M., and Chao, A. (2024). The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey.

[Sapkota et al., 2026] Sapkota, R., Roumeliotis, K. I., and Karkee, M. (2026). Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *Information Fusion*, 126:103599.

[U.S. Securities and Exchange Commission, 2026] U.S. Securities and Exchange Commission (2026). SEC EDGAR Fair Access. https://www.sec.gov/search-filings/edgar-search-assistance/accessing-edgar-data. Accessed: 2026-07-09.

[Xiong et al., 2024] Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. (2024). Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms.

[Zhou et al., 2023] Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., and Chi, E. (2023). Least-to-most prompting enables complex reasoning in large language models.

A Computational Reproducibility

This appendix documents the pipeline execution, evaluation, and data generation commands used to produce the tables presented in this thesis.

A.1 Ground Truth Matching

To match the manifest against the reference dataset and generate the overlap report, run:

```
python scripts/check_overlap.py \
  --manifest manifest.db \
  --gt-dir data/ground_truth \
  --out overlap_report.csv
```

A.2 Pipeline Execution

Initialize the manifest:

```
python run.py discover \
  --start-year 2000 \
  --end-year 2020 \
  --ritter-csv data/IPO-jay_r_ritter.csv \
  --manifest manifest.db
```

Execute the four cells of the experimental matrix separately:

```
python run.py extract \
  --manifest manifest_gen_none.db \
  --gt-overlap-csv overlap_report.csv \
  --extraction-mode generalist \
  --verification-mode none

python run.py extract \
  --manifest manifest_gen_full.db \
  --gt-overlap-csv overlap_report.csv \
  --extraction-mode generalist \
  --verification-mode full

python run.py extract \
  --manifest manifest_spec_none.db \
  --gt-overlap-csv overlap_report.csv \
  --extraction-mode specialist \
  --verification-mode none

python run.py extract \
  --manifest manifest_spec_full.db \
  --gt-overlap-csv overlap_report.csv \
  --extraction-mode specialist \
  --verification-mode full
```

A.3 Evaluation and Table Generation

Generate the common-subset results:

```
python scripts/macro_eval.py \
  --ablation-dir data/ablation_2x2 \
  --bootstrap 1000 \
  --seed 42
```

Generate the end-to-end results:

```
python scripts/macro_eval.py \
  --ablation-dir data/ablation_2x2 \
  --include-failures \
  --bootstrap 1000 \
  --seed 42
```

A.4 Data Summary Commands

Generate the filing-year and status summary:

```
python - <<'PY'
import sqlite3
import pandas as pd

with sqlite3.connect("manifest.db") as con:
    df = pd.read_sql(
        "SELECT first_s1_date, ipo_status FROM companies",
        con,
    )

df["ipo_year"] = pd.to_datetime(
    df["first_s1_date"],
    errors="coerce",
).dt.year

print(pd.crosstab(df["ipo_year"], df["ipo_status"]))
PY
```

Table 10 (Confidence Distribution) telemetry:

```
python3 -c "import json, glob, re; [print(name, '-> Fail:', sum(1
for f in glob.glob(f'path/per_company/*/run_log.json') for t in
json.load(open(f)).get('tool_history', []) if t.get('tool_name') ==
'verify_extraction' and float((re.search(r'conf=([0-9]*?[0-9]+)',
t.get('output_summary', ''))).group(1)) < 0.85)) for name, path in 'Gen':
'data/outputs/2026-07-04_19-08-05', 'Spec':  'data/outputs/2026-07-05_02-48-51'.items()
```

B Reproducibility and Code Availability

The statistical analyses, evaluation scripts, and processed data artifacts presented in this thesis are available at github.com/mmatet/AgenticIPOE. All evaluation scripts are provided in the `scripts/` directory, while the processed CSV exports can be found in `data/evaluation/`.

Note: Raw ground truth data is excluded from this repository due to its confidential nature.

C Agent System Prompts

This section details the system prompts used by the agentic pipeline to orchestrate the extraction and verification process.

C.1 Planner Agent System Prompt

The Planner agent uses the following system prompt to orchestrate tool selection and determine the agent's next status:

```
You are the Planner agent in an agentic IE pipeline for SEC S-1 IPO
   filings.

Your job at each step: examine AgentState and decide the SINGLE next
   action.
You do NOT extract data yourself - you orchestrate tools.

STANDARD FLOW:
  1. fetch_filing       (download S-1 via edgartools)
  2. isolate_section    (extract the Management section)
  3. extract_leadership (LLM extraction)
  4. verify_extraction  (grounded verification)
  5. If verification passes -> RESPOND. If fails -> extract_leadership
     again.

DECISION RULES:
- If a step is already complete (visible in state), do NOT repeat it.
- If verification has passed (verification.passed=true), set next_status
   =RESPOND.
- If verification failed AND extraction_attempts <
   max_extraction_attempts,
  call extract_leadership again (the extractor will see the errors
     automatically).
- If extraction_attempts >= max_extraction_attempts, set next_status=
   RESPOND
  to return best-effort. Never loop indefinitely.
- If a tool failed irrecoverably (e.g. filing not found), set
   next_status=END.

Respond with STRICT JSON only:
```

```
{
  "reasoning": "<one-sentence justification>",
  "next_status": "NEED_TOOL" | "RESPOND" | "END",
  "tool": "<tool_name>" | null,
  "tool_input": { ... }
}
```

C.2 Executor Agent Role

The Executor agent does not rely on a standalone LLM system prompt; instead, it operates via the deterministic logic defined in `executor.py`. Its role is to bridge the Planner's decisions and the Tool Registry by performing the following steps:

- Extracts the tool decision from `state.last_planner_decision`.
- Validates tool applicability via `tool.is_applicable`.
- Executes the tool and records the result.
- Routes the agent status to `RESPOND` if verification passes, or `PLAN` if further action is required.

C.3 Responder Agent Role

Similarly, the Responder agent finalizes the pipeline output through programmatic validation rather than an LLM prompt. Its operational logic is as follows:

- **Validation**: Ensures `state.extracted` contains valid data.
- **Finalization**: Sets `state.final_answer` based on the extraction result.
- **Transition**: Routes the agent to `SUCCESS` if verification passed or if extraction attempts were exhausted (best-effort return).

C.4 Extraction and Verification Prompts

The extraction and verification tools employ specific LLM prompts to ensure grounded and accurate attribute identification.

Listing 1: Generalist Extraction Prompt

```
You are an information-extraction agent for SEC S-1 IPO filings.

Your job: from the Management section provided, extract every named
   executive officer and director, attach a verbatim raw_title, and
   classify each according to the Albert, Eklund, and Tang (2025)
   labelling scheme.
```

```
OUTPUT: A single JSON object {"executives": [...]}, where each executive
    carries name, raw_title, optional bio fields, role_flags (6 booleans
   ), rank_flags (12 booleans), and optional primary_subcategory /
   support_subcategory enums.
```

Listing 2: Specialist Extraction Prompts

```
% Stage 1: Roster
You are a ROSTER-IDENTIFICATION specialist for SEC S-1 IPO filings.
Your ONLY job: from the Management section, list every named executive
   officer AND every named board director. You do NOT classify roles or
   ranks.

% Stage 2: Labelling
You are a LABELLING specialist applying the Albert, Eklund, and Tang (20
   25) scheme to SEC S-1 executives. You are GIVEN a numbered roster and
    source Management section. Your ONLY job: assign role flags, rank
   flags, and subcategories.
```

Listing 3: Full Mode Verification Prompt

```
You are a strict fact-checker for SEC S-1 leadership extractions.

Your job: identify FACTUAL hallucinations and OMISSIONS only. Do NOT
   flag cosmetic differences.

FAIL the extraction when:
  - A person named in the extraction does not appear in the source.
  - An age stated in the extraction is wrong or absent from the source.
  - An education/prior-experience entry references a school or company
     NOT in the source.
  - A clearly-named officer in the source is OMITTED from the extraction
     .

Return strict JSON: {"passed": <bool>, "errors": [<short specific
   strings>], "confidence": <float 0..1>}
```