UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

DÉPARTEMENT DES SCIENCES ADMINISTRATIVES

WORKING PAPER NO. 0001

*Research funded by the FIRC (Fonds institutionnel de recherche et de création), Université du Québec en Outaouais.*

# From Corporate Disclosures to a Russell 3000 Strategic Project Database

*Identification, Measurement, Validation, and Research Applications*

Manel Kammoun  Djerrycharli Tandjambianda  Simon-Pierre Boucher

Département des sciences administratives
Université du Québec en Outaouais

manel.kammoun@uqo.ca · djerrycharli.tandjambianda@uqo.ca · simon-pierre.boucher@uqo.ca

Gatineau – Pavillon Alexandre-Taché
283, boulevard Alexandre-Taché
Gatineau, Québec, Canada J9A 1L8

November 2026

Version 1.1 (revised after peer review; research database under validation)

*Working draft. Author order, affiliations and CRediT contributions remain subject to confirmation by all coauthors (see `docs/contributions.md`). Not for citation without permission.*

**Abstract**

Can strategic projects be identified, classified, dated and linked across the SEC filings of a broad cross-section of U.S.-listed issuers? We document SPID, a project-level research database built for a Russell 3000 proxy (iShares IWV holdings, 2,555 of 2,996 index constituents) merged with the S&P 500: 527,548 Forms 10-K, 10-Q and 8-K of 2,564 registrants filed since 2010, segmented into 1,342,283 sections, of which 703,725 reached a large-language-model extractor in JSON mode. The output is 308,065 section-linked candidate mentions and 121,099 candidate projects under the current key (96,749–198,488 under alternative keys). Mechanical audits find every structured field verbatim in the source section for 88.2% of mentions, a median 9.3% of a 10-K MD&A shown to the model, and 7.3% of mentions describing financing instruments. Precision and recall against human labels are not yet measured; the annotation frame is delivered. SPID is an instrument under validation, not a census of projects.

**Keywords:** SEC filings, text as data, corporate investment, strategic projects, information extraction, large language models, entity resolution, Russell 3000, measurement validation

**JEL Classification:** C55, C81, G31, G38, M41, O33

## 1. Introduction

A large share of what firms do strategically—build plants, open data centers, integrate acquisitions, launch products, move systems to the cloud, commit to decarbonization targets—is described, often repeatedly and over several years, in the narrative sections of their regulatory filings. The standard quantitative record of corporate investment is nevertheless an aggregate: capital expenditure, R&D expense, acquisition spending, or counts of patents and plants. Researchers who want to observe the *unit of decision*—a specific, identifiable initiative with a start, a scope, a budget and a fate—have relied on hand-collected samples of announcements, such as the "Million Dollar Plants" location decisions of Greenstone et al. (2010) or the capital-expenditure announcements studied by McConnell and Muscarella (1985), or have inferred activity indirectly from firm-level text scores (Hoberg and Phillips, 2016; Hassan et al., 2019). This paper documents an attempt to build that unit systematically, from primary sources, for a broad cross-section of U.S.-listed companies, and reports what a set of mechanical audits says about the result.

The object we describe is SPID (*SEC Project Intelligence Database*), developed at the Université du Québec en Outaouais on top of the UQO EDGAR research corpus. Its first release, in June 2026, covered the 500 constituents of the S&P 500 (Tandjambianda et al., 2026). The present paper documents its extension, carried out on October 27–28, 2026, to an issuer universe that approximates the Russell 3000 through the holdings of an index-tracking fund: 2,564 distinct SEC registrants, of which 2,555 come from the iShares IWV holdings file and cover 85.3% of the 2,996 constituents listed in the index factsheet of September 30, 2026 (FTSE Russell, 2026a; BlackRock, Inc., 2026); 527,548 annual, quarterly and current reports filed between January 2010 and October 2026; 1,123,098 filing sections written to an extraction ledger, of which 703,725 passed a lexical gate and were submitted to a large-language-model (LLM) extractor in JSON-object mode; and 308,065 stored project mentions, each linked to its SEC accession number and section identifier, resolved into 121,099 candidate projects under the current resolution key. The extraction was performed in two runs—June 2026 for the S&P 500 block and October 2026 for the rest—under one prompt lineage whose text is unchanged by file-system evidence but for which no June code revision exists (Section 4, Appendix E).

Our research question is methodological: can we reliably identify, classify, date, link and quantify distinct strategic projects disclosed by Russell 3000 companies, while preserving source evidence and without claiming exhaustive coverage of real-world activity? The answer is two-sided and, in this revision, quantified wherever a quantity could be computed without new model calls. On the positive side, the pipeline is re-runnable and ledgered: every submitted section is recorded with its mention count and timestamp, every intermediate artefact is archived with a checksum, and every table, figure and inline number of this paper is regenerated from the exported data by

two scripts. A field-level grounding audit over all 308,065 mentions finds the extracted years, partners and amounts verbatim in the source section in 99.8%, 99.4% and 93.7% of cases, and every structured field of a mention in 88.2% of cases (Section 8). On the cautious side, three results of the same audits bound what the counts mean. First, the focus windows that cap the cost of each call showed the model a median 9.3% of a submitted 10-K MD&A, and 62.7% of all mentions come from sections of which less than a quarter was shown; recall on long sections is therefore bounded by construction, not merely unknown. Second, the project count depends on the resolution key: it ranges from 96,749 to 198,488 across the alternative keys of Section 6, and 54.8% of projects under the current key are grouped on a generic first token of the model's project name. Third, 7.3% of mentions describe share repurchases, credit facilities, note offerings or similar financing instruments that are not projects in the sense of the construct. The components that a validated measurement instrument requires—precision, recall and category accuracy against independent human labels, and an audit of cross-filing linkage—remain unexecuted: the sampling frame, weights and protocol are delivered with this paper, the labels are not. We therefore continue to describe SPID as a *research database under validation*, and we keep a register that separates implemented, measured, documented-but-unverified and proposed claims (Appendix D).

The paper makes four contributions. First, it specifies the measurement target: a *project* is a distinct, identifiable, issuer-associated strategic initiative described in one or more disclosures, as opposed to a *project mention* (a particular sourced statement), a *generic strategy* ("we invest in innovation"), a historical or third-party reference, a financing instrument, or an uncertain candidate (Section 5). Second, it documents an architecture that reads the three main periodic forms for a universe of this size: a frozen issuer universe reconciled against SEC identifiers, an acquisition layer that shares one aggregate rate budget across ten cluster nodes and one external gateway, a parser that keeps section-level provenance, and an extraction ledger that makes the paid inference step resumable (Section 4). Third, it exposes the extraction contract and the resolution rule in enough detail to be audited and, in this revision, audits them: the prompt, the JSON output and its deterministic normalisation, the confidence semantics, the project key, and the ways in which that key both splits and merges real initiatives (Sections 6 and 8). Fourth, it lays out a validation protocol with acceptance gates and reports, gate by gate, what the current build satisfies, what the Stage D audits have measured, and what remains an author task (Section 8). Relative to firm-level text scores (Hoberg and Phillips, 2016; Hassan et al., 2019; Sautner et al., 2023; Jha et al., 2024) and to event-level extraction from 8-Ks (Crane et al., 2024; Dolphin et al., 2026), SPID's unit is the initiative itself, followed across 10-K, 10-Q and 8-K filings; unlike those studies it does not yet report accuracy against human labels, which Section 8 specifies.

Three caveats frame everything that follows. A present-day holdings file of an index-tracking fund that uses representative sampling is a proxy for index membership, not an authoritative

historical constituent list; the universe is a snapshot, and survivorship is built into it (Section 3). A disclosed project is evidence that an issuer *described* an initiative, not evidence that the initiative happened, and a disclosed budget is a statement about intentions or estimates, not realized capital expenditure. Finally, the extractor only sees what the parser and the focus windows give it: sections that were truncated, filings that failed to download or parse, exhibits outside the selected forms, and 10-K items outside the five submitted ones are invisible, and "zero projects" for an issuer-year must never be read as "no projects".

The rest of the paper is organized as follows. Section 2 positions the work relative to text-based measures of corporate activity, LLM-based measurement and extraction, open EDGAR corpora and record linkage. Section 3 defines the universe and reports its reconciliation and its coverage of the index. Section 4 describes acquisition, parsing, the distributed build and the two extraction runs. Section 5 defines the project construct, the gate and focus windows, and the extraction contract as implemented. Section 6 covers mention-to-project resolution, the sensitivity of the project count to the key, timelines and amounts. Section 7 reports descriptive patterns. Section 8 presents the audits executed in this revision, the annotation protocol, and the acceptance gates. Section 9 discusses research applications and identification limits, and Section 10 lists the limitations, the reproducibility statement and the conclusion. The appendices provide the taxonomy, the prompt and gate, the data-model audit, the evidence register, the inter-revision code diff, additional tables and replication instructions.

## 2. Related Literature

**Text as data in accounting and finance.** The narrative content of SEC filings has become a standard empirical input. Loughran and McDonald (2011) showed that general-purpose sentiment dictionaries misclassify financial text and introduced finance-specific word lists; Loughran and McDonald (2016) survey the subsequent decade of textual analysis in accounting and finance, and Gentzkow et al. (2019) provide the general econometric framework for treating text as data, including the measurement-error and validation issues that any text-derived variable inherits. Dyer et al. (2017) document how 10-K disclosure has lengthened and become more repetitive over time, a trend that matters for any extractor that reads the same passages year after year; whether repeated boilerplate produces false-positive projects in SPID has not been measured and is one of the labels of the annotation protocol (Section 8). Cohen et al. (2020) show that changes in otherwise repetitive filing language carry information, which motivates our emphasis on *repeated* mentions of the same initiative across filings rather than on single occurrences. Grimmer and Stewart (2013) offer the general methodological warning that automated text measures must be validated against human judgment on the specific task at hand, which is the stance adopted in Section 8.

**Measuring corporate activity from disclosures.** Closest to our construct are text-based measures of what firms do rather than how they feel. Hoberg and Phillips (2016) build product-market classifications from 10-K business descriptions; Hoberg and Maksimovic (2015) identify financially constrained firms from the language of the liquidity discussion; Hassan et al. (2019) measure firm-level political risk from the share of earnings-call text devoted to politics; Sautner et al. (2023) measure firm-level climate exposure from calls and validate it against green hiring and patenting; Bellstam et al. (2021) score innovation from analyst reports for firms without patents; Li et al. (2021) measure corporate culture with word embeddings; Kelly et al. (2021) measure the novelty of patents from their text, and Kalyani et al. (2025) track the emergence and diffusion of technologies by discovering bigrams in patents and counting them in earnings calls and job postings. Two recent papers obtain firm-level measures directly from a generative model: Kim et al. (2023) summarize earnings-call transcripts into political, climate and AI risk scores, and Jha et al. (2024) prompt a chat model to read managers' investment plans from calls, producing a firm-quarter score that is validated against CFO survey responses and predicts capital expenditure several quarters ahead. SPID differs from all of these in its unit of observation. Rather than a firm-level score summarizing the intensity of a theme, it targets *discrete initiatives*—objects with a name, a type, a status, cited years, a location, partners and an amount—that can be followed from filing to filing. Relative to firm-level text scores (Hoberg and Phillips, 2016; Hassan et al., 2019; Sautner et al., 2023; Jha et al., 2024) and to event-level extraction from 8-Ks (Crane et al., 2024; Dolphin et al., 2026), SPID's unit is the initiative itself, followed across 10-K, 10-Q and 8-K filings; unlike those studies it does not yet report accuracy against human labels, which Section 8 specifies. Table 1 places the database next to the closest prior work on the dimensions that matter for a user: unit, sources, extractor, provenance granularity and the validation reported.

**Table 1.** Positioning of SPID relative to the closest text-based measures, extraction systems and corpora

| Study | Unit | Sources | Method | Provenance | Validation reported |
|---|---|---|---|---|---|
| Hoberg and Phillips (2016) | firm-pair similarity; industry classification | 10-K business descriptions | word-vector similarity | firm-year | against SIC/NAICS classifications on explaining fundamentals |
| Hassan et al. (2019) | firm-quarter political-risk score | earnings calls | pattern-based bigram share | firm-quarter | against volatility, lobbying and donations; human audit of snippets |
| Sautner et al. (2023) | firm-year climate exposure | earnings calls | machine-learning keyword discovery | firm-year | against green jobs and patents |
| Kim et al. (2023) | firm-level risk measures | earnings calls | GPT summarization | firm-call | against volatility and investment outcomes |
| Jha et al. (2024) | firm-quarter investment score | earnings calls | ChatGPT prompting | firm-quarter | against CFO surveys; predicts capex up to nine quarters ahead |
| Crane et al. (2024) | layoff event | Form 8-K | text extraction | filing | against WARN notices (administrative source) |
| Dolphin et al. (2026) | 8-K event, 119-type three-tier taxonomy | Form 8-K | two-stage LLM, labels traceable to the source text | passage | label reliability evaluated |
| Loukas et al. (2021) | item-segmented corpus | all 10-K filings 1993–2020 | HTML parsing, Item segmentation | item | corpus release, no extraction |
| Bettencourt et al. (2026) | layout-faithful corpus | SEC filings, many forms | reconstruction into structured text with tables preserved | document | corpus release, no extraction |
| SPID (this paper) | initiative mention, linked into projects across filings | 10-K, 10-Q, 8-K primary documents 2010–2026, 2,564 issuers | gpt-4o-mini in JSON-object mode behind a lexical gate and focus windows; deterministic key | section (accession, section id); no character span | mechanical grounding of structured fields, 8-K item-code consistency, rules-vs-LLM agreement (Section 8); no human labels yet |

Content of the rows other than SPID is taken from the cited works as summarized in the literature review of this revision; it describes the design of each study, not its results. "Provenance" is the finest unit at which an output can be traced to its source. The row for SPID describes the build at revision `2324b58` after the Stage D audits.

**Large language models as measurement instruments.** The extraction step relies on an instruction-tuned LLM used in a constrained-output mode: a fixed system prompt, a JSON-object response format whose field list is stated in the prompt rather than enforced by a schema, temperature zero, and an explicit instruction to extract only initiatives grounded in the supplied passage. Ludwig et al. (2026) give the econometric framework within which such an output can become a regressor: a measurement generated by a language model is usable for estimation only with a validation subsample that identifies its error structure, and prediction uses require that the model's training data not contain the outcome. SPID currently has the first half of that framework—the

instrument and its ledger—and not the second; Section 8 is written as the specification of that validation subsample. de Kok (2025) sets out the practical protocol for generative models in accounting research (prompt and model versioning, temperature, repeated runs, validation), several items of which the current build does not meet: no prompt hash or model snapshot is recorded per run and the June model snapshot is unknown (Section 4). Gilardi et al. (2023) show that zero-shot LLM annotation can match or exceed crowd workers on text-classification tasks, which is the comparison the human-versus-model benchmark of Section 8 is designed to make for this construct; Afzal et al. (2026) report accuracy for a generic LLM framework extracting structured fields from narrative financial reports. Closest in design are Dolphin et al. (2026), who label 8-K disclosures against a fine-grained taxonomy with labels constrained to be traceable to the source text and evaluate their reliability, and Crane et al. (2024), who extract one event type from 8-Ks and validate it against an administrative source; both report the reliability that SPID does not yet have. The model is asked to prefer fewer, concrete projects to many vague ones, to return an empty list when no concrete project is described, and to attach a self-reported confidence. We make no claim that this confidence is calibrated: verbalized confidence scores of language models are known to be poorly calibrated (Yang et al., 2024; Ye et al., 2024), a missing confidence is defaulted to a fixed value by the post-processing (Section 5), and calibration is one of the items of the validation protocol. Semantic search over projects uses sentence embeddings (Reimers and Gurevych, 2019).

**Open EDGAR corpora.** The corpus layer of SPID re-implements a step that prior work has already published: EDGAR-CORPUS (Loukas et al., 2021) provides every 10-K from 1993 to 2020 segmented by Item, and the Stanford EDGAR Filings Dataset (Bettencourt et al., 2026) reconstructs filings into layout-faithful text in which tables are preserved. SPID's parser covers 10-K, 10-Q and 8-K primary documents from 2010 to 2026 for its universe, flattens tables to tab-separated text and stores no character offsets into the raw HTML (Section 4); for the S&P 500 block the 10-K items were prepared by an earlier parser that discarded tables altogether (Section 4). A user who needs table-level fidelity should prefer those corpora as the text source and use SPID for its project layer.

**Entity resolution and record linkage.** Linking repeated mentions of the same initiative across years is a record-linkage problem in the sense of Fellegi and Sunter (1969); Christen (2012) reviews the modern toolbox of blocking, comparison and classification. The current SPID resolver is a deterministic blocking rule (issuer, project type and a location or name token) rather than a probabilistic linker, and Section 6 shows how much the project count depends on it. The migration to an explainable, reversible linkage with match scores and human adjudication of ambiguous clusters is part of the roadmap; pre-trained language models have been shown to improve entity matching over string-similarity baselines (Li et al., 2020), and generative models can be used for

the same task (Peeters et al., 2023), which is the direction the proposed `project_mention_links` table is designed to accommodate.

**Infrastructure.** The database is implemented as a single analytical file in DuckDB (Raasveldt and Mühleisen, 2019), with Parquet exports; acquisition respects the SEC's published fair-access policy (U.S. Securities and Exchange Commission, 2026), and the Russell 3000 is defined following FTSE Russell's published construction methodology (FTSE Russell, 2026b), approximated here through a fund-holdings proxy whose coverage of the index is quantified in Section 3.

## 3. Sample: Issuer Universe and Filing Coverage

### 3.1. Estimand: current constituents of a proxy at a snapshot date

Two different populations can be labeled "the Russell 3000". The first is the set of *historical point-in-time constituents*: for each date, the companies that FTSE Russell included in the index after its reconstitution (FTSE Russell, 2026b). The second is the set of *current constituents at a snapshot date*. The two differ because of index entries and exits, mergers, delistings and IPOs; a panel built on the second population is subject to survivorship and backfilling biases that a point-in-time panel is not. SPID targets the second population, and it observes it through a proxy. No licensed historical constituent list was available to the team at the time of the build, and FTSE Russell does not publish machine-readable membership lists for free; the universe is therefore read from the holdings file of the iShares Russell 3000 ETF (IWV). The index factsheet of September 30, 2026 lists 2,996 constituents (FTSE Russell, 2026a); the IWV holdings of October 23, 2026 map to 2,555 SEC registrants, 85.3% of that count. The fund's prospectus states that it uses a representative-sampling strategy rather than full replication (BlackRock, Inc., 2026), so it need not hold every index member, and which members it omits cannot be determined from public documents; the gap of at most 441 securities (14.7% of the index count) is an upper bound on omitted issuers, and under sampling the omitted names are expected to be the smallest, that is, the extension beyond the S&P 500 that motivated the build. The three dates differ (index characteristics September 30, 2026, holdings October 23, 2026, download October 27, 2026), and the index counts securities while the proxy counts issuers after collapsing share classes, so issuer-level coverage is somewhat higher than the printed ratio. Any inference about the "Russell 3000 over 2010–2026" from this database is therefore an inference about firms that were in the IWV proxy in October 2026 and about their filings back to 2010.

### 3.2. Universe construction and reconciliation

The issuer universe is frozen in a single JSON artefact, `data/metadata/universe.json`, built by `scripts/build_universe.py` and documented in `docs/universe_definition.md`; the artefact and its two inputs are archived with the paper with SHA-256 checksums (`data/metadata/SHA256SUMS`; digests in Table 25). It merges two sources on the SEC Central Index Key (CIK):

1. the S&P 500 constituent list scraped from the public Wikipedia table on May 25, 2026 (503 rows, 500 distinct CIKs, with GICS sector and sub-industry) (Wikipedia contributors, 2026), used for the first SPID release; and
2. the holdings file of the iShares Russell 3000 ETF published by iShares as `latest-holdings.csv`, with holdings as of October 23, 2026, downloaded on October 27, 2026 (BlackRock, Inc., 2026). The file carries the fund's own sector labels.

Tickers are normalized to the SEC convention (share-class separators become hyphens: "BRK B" $\rightarrow$ `BRK-B`) and mapped to CIKs through the SEC's public `company_tickers.json` (10,430 entries at fetch time). Table 2 gives the reconciliation from input securities to issuers with resolved projects, and two of its steps must be read with care. The first line, 2,587 equity lines with a ticker, is the count reported by the build log: neither the downloaded holdings file nor the ticker file was archived, the build script drops non-company lines (contingent value rights, escrow shares, rights when-issued) without recording them, and the difference between that count and the 2,573 mapped lines is therefore not reconstructible line by line. The 7 tickers that reached the mapping step without a CIK are listed in the artefact's `unmatched` field: four banks that file with the FDIC rather than the SEC and three tickers absent from the SEC ticker file on the download date, for which no reason is recorded. Earlier project documents gave other counts of excluded lines; the artefact count of 7 is the only one that can be verified and is the one used here. The 2,573 mapped lines collapse to 2,555 CIKs because 18 registrants appear with two share classes. The merged universe of 2,564 issuers is 500 S&P 500 CIKs plus 2,555 proxy CIKs minus 491 present in both lists; 9 issuers are in the S&P 500 snapshot but no longer in the IWV file (typically recently acquired) and are retained with an `in_sp500` flag.

**Table 2.** Universe-to-dataset reconciliation: from the two constituent lists to issuers with resolved projects

| Step | Count |
|---|---|
| iShares IWV holdings file, equity lines with a ticker (as of Oct 23, 2026); count reported by the build log, source file not archived | 2,587 |
| of which mapped to an SEC CIK (CVRs, escrow lines and FDIC-only banks excluded) | 2,573 |
| distinct CIKs (share classes collapsed) | 2,555 |
| S&P 500 constituents, cached snapshot of 2026-05-25 (500 CIKs) | 503 |
| Merged issuer universe (distinct CIKs) | 2,564 |
| in both lists | 491 |
| S&P 500 only | 9 |
| Russell proxy only | 2,064 |
| Russell 3000 index constituents, FTSE Russell factsheet as of 2026-09-30 | 2,996 |
| coverage of the index by the proxy (distinct CIKs / constituents, %) | 85.3 |
| Issuers with at least one 10-K, 10-Q or 8-K since 2010 | 2,499 |
| Issuers without any such filing (foreign private issuers filing 20-F/40-F, FDIC-only banks) | 65 |
| Issuers with at least one resolved project | 2,492 |

Sources: `data/metadata/universe.json`, `russell3000_constituents.json`, `sp500_constituents.json` (SHA-256 prefixes: `universe.json` 3967b29b378e…; `russell3000_constituents.json` 758cbebf5854…; `sp500_constituents.json` 508407526069…; full digests in Table 25) and the merged DuckDB database. The first row is the count reported by the build log of `scripts/build_universe.py`; the iShares holdings file and the SEC `company_tickers.json` used by the build were not archived, so this count and the dropped non-company lines are not reconstructible (Section 3). The mapped lines collapse to fewer CIKs because 18 dual-class registrants appear twice; 7 tickers have no CIK in the SEC map (TOWN, FRBA, NBN, HIFS, ADRO, THRD, HOLX). The index count is the "Number of Holdings" of the FTSE Russell factsheet (SHA-256 281e92643781…); the fund tracks the index by representative sampling, so the omitted constituents cannot be identified without the licensed constituent list. The 65 issuers without filings are listed in `data/issuers_without_filings.csv` (foreign private issuers filing Forms 20-F/40-F and banks filing with the FDIC); 7 issuers have filings but no resolved project.

Several design decisions deserve to be stated plainly. The proxy universe counts *issuers*, not securities. The universe table records, for each issuer, the SEC name, SIC code and description, state of incorporation, exchange and fiscal-year end from the SEC submissions header, the fund weight, and the cluster node that processed its filings (`shard_node`). The research-design specification

against which the build is audited in `docs/implementation_audit.md` (hereafter "the specification": the list of required tables, keys and validation gates for a research-ready release) calls for a `universe_snapshots` table with source checksums, membership dates and match confidence; this is only partially met. Snapshot date, source URL and match method are recorded and the derived artefacts are now hashed, but the source files are not archived and no per-row match confidence is stored, which is why gate (a) of Table 20 is reported as partially met.

The June 2026 release remains a subset of the October build in the sense that every S&P 500 issuer is in the universe, but not in the sense of identical counts. The June block yielded 19,227 projects from 39,930 mentions, both reproduced exactly when the resolver is run on the backfilled rows alone; the same 500 issuers now carry 27,766 projects from 70,367 mentions. The increase is explained by scope, not by the resolver: the June run had read only 3,710 S&P 500 10-Qs (23,873 sections, 5,817 mentions), and the October run added 116,231 10-Q sections of 19,399 filings for the same issuers (28,963 mentions) plus 910 10-K and 564 8-K mentions. Of the 27,766 S&P 500 projects, 15,910 rest on June mentions only, 3,317 on both runs and 8,539 on October mentions only (Appendix E).

### 3.3. Filing scope

For every issuer the pipeline enumerates the SEC submissions index (`data.sec.gov/submissions/CIK{cik}.json`, including the paginated archive blocks) and retains the primary document of every Form 10-K, 10-Q and 8-K filed on or after January 1, 2010. Amendments (`/A`), exhibits other than the primary document, proxy statements, registration statements and foreign-issuer forms are out of scope. Within the retained documents, the extractor reads Items 1, 1A, 2, 7 and 7A of the 10-K, all seven segmented items of the 10-Q (Part I Items 1–4, Part II Items 1, 1A and 2) and the body of the 8-K; 10-K Items 3–6 and 8–9B are parsed but not submitted, and a 10-K in which the parser recognizes no Item heading at all is kept as one `full_document` section that is outside the submitted scope (86 filings, of which 51 were covered by the earlier S&P 500 ingestion and 35 are invisible to SPID; Section 4). The choice reflects where strategic initiatives are described in prose, and it is a defensible but not innocuous restriction: press releases attached as Exhibit 99 to an 8-K, which often carry the first and most detailed announcement of a project, are not read when they are not the primary document. Testing whether this scope omits important announcements is part of the validation roadmap.

**Table 3.** Filings in the corpus by form, 2010-01-04 to 2026-10-27

| Form | Filings | First filing | Last filing | Raw HTML (GB) | Parser error | Missing on disk |
|---|---|---|---|---|---|---|
| 10-K | 31,775 | 2010-01-08 | 2026-10-27 | 88.6 | 2 | 1 |
| 10-Q | 95,167 | 2010-01-05 | 2026-10-27 | 155.9 | 11 | 1 |
| 8-K | 400,606 | 2010-01-04 | 2026-10-27 | 10.0 | 1 | 0 |
| Total | 527,548 | 2010-01-04 | 2026-10-27 | 254.5 | 14 | 2 |

Primary documents of Forms 10-K, 10-Q and 8-K retained from the SEC submissions index of every issuer of Table 2. "Parser error" counts documents whose primary document could not be cleaned; "missing on disk" counts documents logged as downloaded but not found at parse time. Raw HTML size is the sum of primary-document sizes recorded at download time. The 14 parser errors and 2 documents logged as downloaded but absent on disk are listed by accession number in `results/universe/filings_status_missing.csv` (parser errors: BJRI, NEWT, NNN, UNP; missing: XEL 10-K 2014, LNT 10-Q 2012).

Table 3 summarizes the corpus: 527,548 primary documents, 254.5 GB of raw HTML, of which 16 documents (0.003%) were not parsed. Figure 1 and Table 26 in the appendix show the coverage by year. The number of issuers with at least one filing rises from 1,392 in 2010 to 2,498 in 2026: the snapshot universe contains many companies that went public after 2010, so the early years of the panel are populated by the older and larger survivors. This is the concrete form taken by the survivorship caveat above.

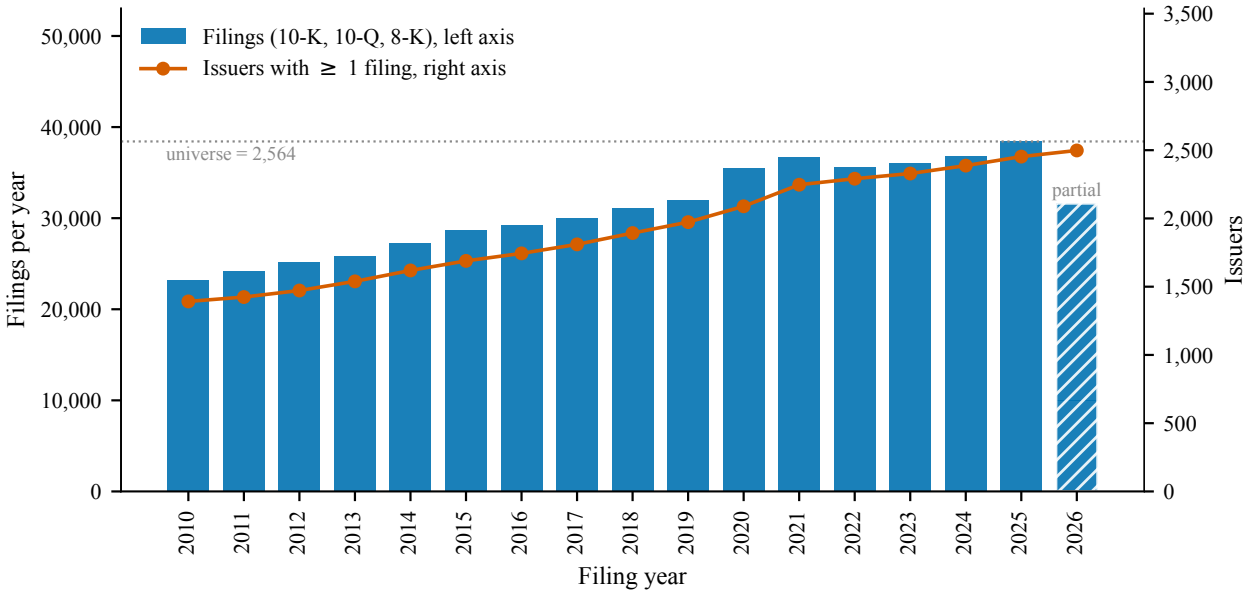



**Figure 1.** Filings and issuers per year in the corpus, 2010–2026. Bars: primary documents of Forms 10-K, 10-Q and 8-K by filing year (left axis; the hatched 2026 bar covers January 1 to 2026-10-27). Line: issuers with at least one filing in the year (right axis), against the 2,564 issuers of the snapshot universe (dotted). Issuers that went public after 2010 enter the panel at their first filing, so the number of observed issuers rises mechanically under the snapshot design (Section 3); counts are exact, no sampling uncertainty applies.

# 4. Architecture: Acquisition, Parsing and the Distributed Build

## 4.1. Overview

Figure 2 summarizes the pipeline. Two layers are distinguished throughout. The *corpus layer* (code in `src/` and `scripts/`) acquires and parses filings, chunks them, embeds the chunks and computes thematic factor scores; it is universe-agnostic and produces a DuckDB database (`edgar_research.duckdb`, 68 GB) plus Parquet exports (29.1 GB of section text). The *project layer* (code in `project-db/spid/`) reads the Parquet exports through views, runs candidate gating and LLM extraction, resolves mentions into projects, builds a knowledge graph and project embeddings, and persists everything in a second, much smaller DuckDB file (`spid.duckdb`, 3.2 GB). The separation matters for re-runs: the paid step (LLM extraction) depends only on the frozen Parquet exports and on a ledger of processed sections, so it can be resumed or audited independently of the corpus build. One exception to the separation, documented in Section 4.3, is that the 10-K items of the S&P 500 block were prepared in June 2026 by a parser internal to the project layer and are reused as such.

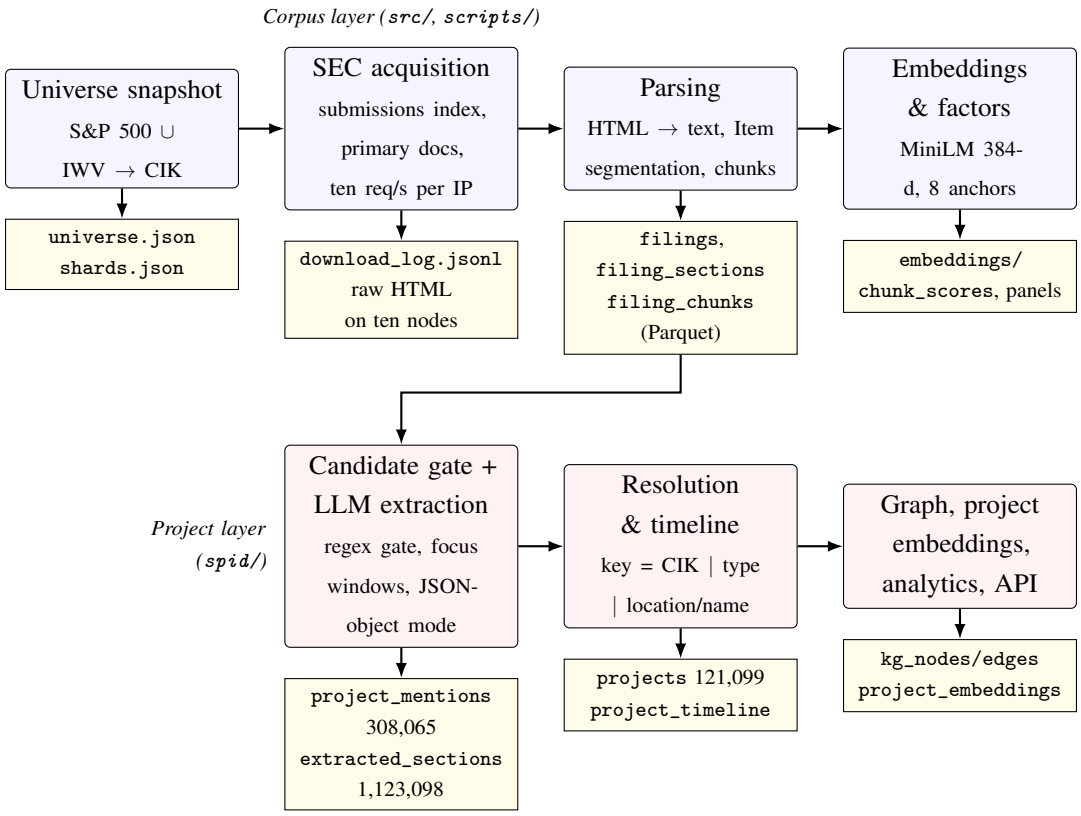



**Figure 2.** Pipeline architecture. Rounded boxes are processing steps; rectangles are persisted artefacts. The project layer reads the Parquet exports of the corpus layer through read-only views and never modifies them; the 10-K items of the S&P 500 block are the exception described in Section 4.3. The diagram is drawn by hand (TikZ), not generated from data; the counts shown are the inline macros of `tables/numbers.tex`.

### 4.2. SEC acquisition

The SEC's fair-access policy allows ten requests per second per client and asks for a descriptive `User-Agent` with a contact address (U.S. Securities and Exchange Commission, 2026). Both constraints shaped the acquisition layer (`src/acquisition/bulk_downloader.py`). Requests are issued by a pool of threads behind a single token bucket; the bucket rate is a parameter, and when several processes share one public IP address the orchestrator divides nine requests per second among them. HTTP 403, 429 and 503 responses trigger a 65-second cool-down for the whole process rather than a per-request retry, so that a throttling signal is never answered with a burst. Downloads are idempotent: a filing is skipped when its file exists, and every completed download is appended to a JSON-lines log carrying the accession number, form, filing and report dates, primary document name, size and local path. A trusted-log option lets a node skip filings that an earlier corpus already contains without re-fetching them. The SEC submissions header of each issuer (name, SIC, state of incorporation, exchanges, fiscal-year end, former names) is saved alongside, and feeds the `companies` table.

The October 2026 build used eleven machines: ten Apple-silicon nodes of the UQO MacLustr cluster behind one residential IP address, and one Linux server in an OVHcloud data center in Beauharnois, Québec, behind a second address. Issuers were assigned to nodes in proportion to CPU cores; half of each node's issuers were downloaded by the node itself (0.9 requests per second each) and half by the external server (one request per second per shard), then shipped to the node over SSH. A first allocation by contiguous CIK ranges proved badly unbalanced—old CIKs are large filers with many documents each—and was replaced after one hour by an interleaved allocation of the not-yet-started issuers. According to the run notes of the research log, the acquisition of about 400,000 new documents took seven hours at a sustained 17 requests per second with no fair-access violation recorded; the orchestrator logs behind these figures are not archived with the paper, so they are reported as run facts rather than as exported statistics. Two filings are logged as downloaded but were not found on disk at parse time (Table 3); in substance they are download failures.

### 4.3. Document processing: two parsing paths

The corpus layer cleans each primary document with `lxml`: `script`, `style`, `meta` and `link` elements are removed, tables are flattened to tab-separated text, lists are kept as line items, whitespace is normalized and files above 50 MB are skipped. The segmenter then locates "Item *n*" and "Part I/II" headings by regular expression: 10-Ks are split into up to fourteen sections (Items 1 to 9B), 10-Qs into seven (Part I Items 1–4, Part II Items 1, 1A, 2), and 8-Ks are kept as a single `full_body` section. A filing in which no heading is found is kept as one `full_document` section rather than dropped, so that the foreign-key chain from chunks to sections to filings is never broken; for 10-Ks such sections lie outside the submitted scope (86 filings), for 10-Qs they are submitted (241 of 243). Sections are chunked into 512-word windows with a 64-word overlap for the embedding and factor layers (10,720,890 chunks). Provenance is preserved at the section level: each section row stores its item number, order, word and character counts, and each chunk stores its character offsets within the section. Character-level offsets of sections within the raw HTML are *not* stored, a limitation recorded in the audit.

The 10-K items of the S&P 500 block did not go through this path. They were produced in June 2026 by the project layer's own ingestion module (`spid/ingest.py`: BeautifulSoup, every `<table>` element removed, a separate Item regular expression, the longest instance of each Item kept) and stored in a legacy table (`spid_sections`, 27,164 rows for 7,368 filings); the October project layer keeps the June text for every 10-K already present in that table and reads the corpus-layer sections only for the others. The two segmenters produce measurably different text. On the 25,673 item pairs available in both tables, the legacy MD&A has a median of 9,921 words against 11,661 for the same item in the corpus layer, 141 dollar signs on average against 272, and 596 numeric tokens against 1,177; 92.3% of MD&A pairs carry fewer money tokens in the

legacy text, and the legacy Properties item is less than half the length of its counterpart in 19.4% of pairs (`results/runs/segmenter_item_comparison.csv`). Capital-expenditure tables, which the legacy ingestion removed, are among the passages a capex extractor would most want to see. The 1,230 rows labeled `full_body` in Table 10 are also legacy rows: whole 10-Ks (median 44,514 words) in which the June ingestion found no Item heading, squeezed through the focus windows of Section 5; the corpus layer has proper items for 1,117 of these filings, but those 5,385 in-scope sections were never submitted because the project layer defers to the legacy table. The earlier version of this paper attributed the 1,230 rows to the October segmenter; that was wrong. The practical consequence is that the S&P 500 cohort's 10-K text and the Russell-only cohort's 10-K text were prepared by different cleaners, which Section 7 and Appendix E take into account when the two cohorts are compared.

Table 4 reports the outcome by form. Two coverage statistics bound what the extractor could see: 0.003% of filings were not parsed, and 35 10-K filings have neither recognized items nor a legacy row and are invisible to the project layer. The 10-K funnel reconciles exactly: 341,207 parsed 10-K rows minus 188,518 out-of-scope items, minus 35,037 rows of filings covered by the legacy table, minus 1,112 rows under 30 words, gives 116,540 corpus-layer rows, which with the 27,164 legacy rows equal the 143,704 10-K rows of the ledger.

**Table 4.** Parsed sections, extraction ledger, gate-passing calls and mentions, by form

| Form | Parsed sections | Median words | Ledgered rows ($\geq$ 30 words) | Gate-passing calls | Sections with $\geq$ 1 mention | Mentions returned (ledger, before de-dup.) | Mentions stored | Issuers ledgered |
|---|---|---|---|---|---|---|---|---|
| 10-K | 341,207 | 875 | 143,704 | 135,310 | 30,877 | 81,622 | 81,495 | 2,464 |
| 10-Q | 600,471 | 500 | 578,793 | 363,973 | 75,705 | 152,261 | 152,004 | 2,494 |
| 8-K | 400,605 | 575 | 400,601 | 204,442 | 61,648 | 74,594 | 74,566 | 2,499 |
| Total | 1,342,283 | | 1,123,098 | 703,725 | 168,230 | 308,477 | 308,065 | 2,499[†] |

"Ledgered rows" counts rows of the `extracted_sections` ledger (parsed sections with at least 30 words; for 10-Ks restricted to the items of Table 10); the ledger records neither whether the candidate gate passed nor whether the API call succeeded, so it is not a count of calls. "Gate-passing calls" recomputes the candidate gate of Appendix B over the ledgered text: 703,725 sections reached the model and 419,361 were ledgered with zero mentions without a call (Table 7 splits them by run). "Mentions returned" is $\sum$`n_mentions` over the ledger (308,477); "Mentions stored" is the row count of `project_mentions` (308,065). The difference of 412 (10-K 127, 10-Q 257, 8-K 28), spread over 304 sections, is the number of mentions overwritten before storage because the mention identifier hashes the type and the first 40 characters of the project name (Section 5); every count in the text uses the stored basis unless labelled "ledger". [†]Issuers with at least one filing of any form, not a column sum.

### 4.4. Distributed build and merge

Parsing, chunking and embedding ran on every node against the node's own shard, using the same code revision and a local DuckDB file; each node then exported its tables to Parquet. The master node (M3U96b, which also holds the original S&P 500 corpus) pulled the shards and merged them with an append-only rule: a row is inserted only if its primary key (accession number, section or chunk UUID) is absent, so that re-running the merge cannot create duplicates. The merged database holds 2,564 issuers, 527,548 filings, 1,342,283 sections, 10,720,890 chunks and as many 384-dimensional embeddings. The run notes record a parsing throughput of about 100 filings per second per node (five to thirteen minutes per shard), 15–40 minutes per node for the embeddings on Apple-silicon GPUs, and seven minutes for the merge and the factor computation; like the acquisition figures these are run facts, not exported statistics. The orchestration script (`scripts/cluster/orchestrate.py`) is stage-based and idempotent: every stage polls marker files on the nodes and can be restarted without side effects. Private infrastructure details (addresses, credentials, keys) are kept outside the repository and outside this paper.

### 4.5. Versioning and the two extraction runs

The repository is versioned on a private Git server. The revision used for the October 2026 build is `2324b58` (October 28, 2026). The repository holds six commits, all dated October 27–28, 2026, so no revision of the code exists for the June 2026 run; the earlier version of this paper and the June methodological study (Tandjambianda et al., 2026) cited `df7905d` as the June revision, but that commit is dated October 27, 2026. What can be established is file-system evidence: the extractor modules (`llm.py`, `rules.py`, `base.py`, `taxonomy.py`, `resolve.py`, `ingest.py`) carry modification times of June 17, 2026, before the June run, while the driver files (`config.py`, `pipeline.py`, `store.py`) were edited on October 27, 2026 before the first commit, so their June versions are unrecoverable. The diff between the two October revisions on the extractor, parser and driver files is $+3/-1$ lines in `store.py` (a DuckDB 2.0 Arrow fallback), and the diff from the first commit adds $+13/-7$ lines in `filing_parser.py`; both patches are reproduced in Appendix E. The prompt text is therefore unchanged across the two runs as far as the file system can show, which is weaker than a per-run digest.

The extraction ledger does not help here. Its `backend` column holds the constant string `llm` for every row of both runs; neither a model identifier nor a prompt, schema or taxonomy hash is stored, and the run logs contain no model string either. The October run is reported by the authors as having used the snapshot `gpt-4o-mini-2024-07-18`, and the June run the unpinned alias `gpt-4o-mini`; the first identity rests on the authors' record rather than on an archived artefact, and the June snapshot is unknown. The June block was not re-extracted in October: its 154,964

ledger rows were written at a single timestamp (October 27, 2026, 13:12) when the ledger table was created, without the 30-word filter that the October run applied (1,070 rows are shorter), which is why the ledger's time range begins on October 27 and why the two runs can be separated only by that timestamp. The proposed `extraction_runs` table (Appendix C) records, per run, the model snapshot, the `seed` and `system_fingerprint` returned by the API, and the prompt and taxonomy hashes; none of these exists for the current build.

## 5. The Project Construct and the Extraction Contract

### 5.1. What counts as a project

The unit of measurement is a *distinct, identifiable, issuer-associated strategic initiative* involving a planned, announced, ongoing, completed, postponed or cancelled action: a plant expansion, a research program, a technology deployment, an AI initiative, a data-center build, a supply-chain transformation, an acquisition and its integration, a renewable-energy investment, and the other categories of the taxonomy. The database stores two tiers and, in its intended final form, six labels:

- `project_mention`: a particular sourced statement about an initiative, in one section of one filing, with the extracted fields and the model's one-sentence description;
- `project`: a resolved initiative described by one or more mentions, carrying a stable identifier and a timeline;
- `generic_strategy`: non-specific aspirations ("we continue to invest in innovation", "we may pursue acquisitions") that the contract instructs the model to exclude and that the resolver never sees unless the model fails to exclude them;
- `historical_or_third_party_reference`: initiatives of a competitor, a customer or a former parent, mentioned in the issuer's filing but not attributable to the issuer;
- `financing_instrument`: share-repurchase programs, dividends, credit facilities, term loans, note offerings, at-the-market programs and commercial paper, which are corporate actions but not projects in the sense above; the taxonomy has no category for them, the gate word "program" admits them, and the prompt's "any concrete initiative, investment, program" does not exclude them;
- `uncertain_candidate`: a plausible initiative without enough identity evidence (no name, location, counterparty or amount) to be counted.

The current build implements the first two tiers directly. The third is enforced only through the prompt and the confidence threshold; the fourth, fifth and sixth exist as instructions and as a confidence score, not as stored labels. The financing class was added in this revision after a regular-expression pass over the stored mentions (`repurchase|buy-?back|dividend|credit facility|revolving|term loan|notes? (due|offering)|at-the-market|senior`

notes|commercial paper on the project name and description) matched 22,412 mentions, 7.3% of the total, carrying 17.8 T of the 104.2 T USD of mention amounts; the match concentrates in Part II Item 2 of the 10-Q (54.5% of its mentions, where issuer repurchases are tabled) and in 8-K bodies (19.2%), and by category in capex_program (52.6% of its mentions), cost_reduction (16.4%) and ma_integration (7.0%). Section 7 reports the category counts with and without this class, and Table 5 gives the match by item, category and term. Promoting the labels to explicit columns of the mention table, filled by annotators and eventually by a classifier, is the first schema change proposed in docs/project_ontology.md.

**Table 5.** Financing-instrument, buyback and dividend mentions: a construct-violation class detectable by pattern

| *Panel A: by form and item* | | | | | *Panel B: by category* | | | |
|---|---|---|---|---|---|---|---|---|
| Form | Item | Mentions | Matched | % | Category | Mentions | Matched | % |
| 10-K | md_and_a | 39,021 | 509 | 1.3 | Capital expenditure program | 19,211 | 10,096 | 52.6 |
| 10-K | business | 30,455 | 116 | 0.4 | M&A / integration | 130,259 | 9,145 | 7.0 |
| 10-K | properties | 7,869 | 4 | 0.1 | Cost reduction / restructuring | 13,157 | 2,160 | 16.4 |
| 10-K | risk_factors | 2,796 | 14 | 0.5 | Strategic partnership | 17,501 | 475 | 2.7 |
| 10-K | full_body | 824 | 2 | 0.2 | Technology deployment | 6,725 | 360 | 5.4 |
| 10-K | quantitative_disclosures | 530 | 37 | 7.0 | Plant / site construction | 18,388 | 40 | 0.2 |
| 10-Q | md_and_a | 99,167 | 1,372 | 1.4 | Infrastructure investment | 8,510 | 37 | 0.4 |
| 10-Q | financial_statements | 30,080 | 721 | 2.4 | Sustainability / ESG | 2,607 | 36 | 1.4 |
| 10-Q | unregistered_sales | 9,506 | 5,179 | 54.5 | Geographic expansion | 17,713 | 27 | 0.2 |
| 10-Q | controls_and_procedures | 7,079 | 11 | 0.2 | Manufacturing / capacity expansion | 8,825 | 13 | 0.2 |
| 10-Q | risk_factors | 4,420 | 43 | 1.0 | R&D program | 11,758 | 7 | 0.1 |
| 10-Q | legal_proceedings | 845 | 8 | 1.0 | Data center | 889 | 6 | 0.7 |
| 10-Q | quantitative_disclosures | 800 | 76 | 9.5 | Drug / clinical pipeline | 26,755 | 5 | 0.0 |
| 10-Q | full_document | 107 | 0 | 0.0 | New product / launch | 12,456 | 3 | 0.0 |
| 8-K | full_body | 74,566 | 14,320 | 19.2 | Energy transition / renewables | 866 | 2 | 0.2 |
| All | | 308,065 | 22,412 | 7.3 | All | 308,065 | 22,412 | 7.3 |

A mention is matched when its lower-cased name or description contains repurchase|buy-?back|dividend|credit facility|revolving|term loan|notes? (due|offering)|at-the-market|senior notes|commercial paper (most frequent terms: repurchase 10,886, senior notes 3,375, term loan 2,263, revolving 1,839, credit facility 1,184, notes due 1,177). These instruments are not strategic projects under the construct of Section 5, yet the gate word "program" and the prompt admit them; 13,139 projects (10.8%) have at least one matched mention and 12,027 a matched representative name or description. Matched mentions carry 17.8 of the 104.2 T USD of mention amounts. The pattern is a lower bound on the class (no human label has confirmed any match or non-match) and is applied to the stored paraphrase, not to the filing text. Part I/Part II "Item 2" collisions of the 10-Q segmenter affect 1,001 unregistered_sales sections above 3,000 words (1.3% of 77,869 pairs), 296 of them with an MD&A sibling under 500 words.

### 5.2. Taxonomy

The SPID taxonomy has 21 categories (Table 21 in Appendix A), defined in `spid/taxonomy.py` with a human-readable label and a set of lexical signals used by the rule-based extractor and by the candidate gate. The LLM extractor receives the 21 labels verbatim in its system prompt and must choose exactly one per project. A label outside the list is mapped back to a category by substring matching in a fixed order and, failing that, to `technology_deployment`; an unrecognized status is mapped to `mentioned`. Neither fallback is logged, and the raw model output is not stored, so their frequency can only be bounded from the stored fields: 90 of the 124 mentions that carry both default values have no lexical signal of any category in their name or description, a lower bound on silently defaulted mentions, and the lexical signal of the assigned category is present in the name or description of 79.8% of `ma_integration` mentions but only 5.4% of `technology_deployment` mentions (`results/grounding/fallback_bounds.json`). The categories are not mutually exclusive in substance (an acquired plant is both `ma_integration` and `plant_construction`; a data center is also `infrastructure`), and the model's choice in such cases is a classification decision we have not yet audited against human labels. We document the taxonomy as it stands rather than redesign it: any extension would be versioned, with a crosswalk from old to new codes, so that counts published under the current version remain reproducible.

### 5.3. Candidate gating and focus windows

A section is written to the extraction ledger if it contains at least 30 words; it reaches the model only if it also matches a broad lexical gate (the case-insensitive regular expression reproduced from the code in Appendix B: project, initiative, program, invest, expan-, construct, build, launch, develop, deploy, implement, acqui-, partner, facilit-, capacity, capital, migrat-, transform, pipeline, plant, data center, automat-, sustainab-, renewable, restructur-). The gate is applied inside the extractor, which returns an empty list *before any API call* when it fails, and the pipeline writes that result to the ledger as "extracted with zero mentions" exactly as it writes a model abstention. The ledger therefore counts 1,123,098 rows but only 703,725 calls; 419,361 rows (37.3%) were never sent, and no section with a stored mention fails the gate. The gate is deliberately permissive—it exists to avoid paying for sections that cannot contain a project—and its false-negative rate cannot be estimated from the ledger because gate rejections and abstentions are recorded identically; the annotation frame of Section 8 draws gate-rejected sections explicitly for that purpose.

For sections longer than 9,000 characters, the extractor collects windows of 220 characters before and 380 characters after each gate hit, in document order, merging overlaps, stops collecting at 12,600 characters of source text, emits at most 9,000 characters, and cuts the user message at 14,000 characters (a limit never reached: 0 messages were cut). The function is deterministic, so

what the model was shown can be reconstructed for every ledgered section without a model call, and this revision does so (`results/focus/`; Table 6, Figure 3). 326,095 ledgered sections (29%) and 324,386 of the 703,725 calls (46%) exceed the budget; for them the mean fraction of the text shown is 15.9%. The submitted 10-K MD&A sections (28,919 of them) have a median length of 9,905 words or 64,194 characters; the model saw a median 9.3% of each (tenth percentile 4.3%, ninetieth 24%), the last character shown lies at 16% of the text in the median section, 90.5% of these sections were shown for less than a quarter, and the median share of gate hits inside the shown windows is 15.6%. The medians are 15.7% for the 10-K business item, 10.1% for risk factors, 13.2% for the 10-Q MD&A and 10.6% for the 10-Q financial-statement notes, whereas 91% of 8-K bodies are shown whole. Because the windows are taken in document order and the budget is reached early, the liquidity and capital-resources discussion that closes a long MD&A is structurally under-represented: only 5.8% of the occurrences of "capital expenditure", "liquidity and capital resources" or "commitment" in over-budget 10-K MD&As fall inside a shown window (10.0% in the business item, 8.3% in the 10-Q MD&A).

Two consequences follow. First, 193,416 of the 308,477 mentions returned by the model (62.7%) come from sections of which less than a quarter was shown—84% of 10-K mentions, 79% of 10-Q mentions and 6% of 8-K mentions—so most of the database rests on partial readings of its sources. Second, the yield is bounded by the budget rather than by the text: the number of mentions per 10-K MD&A rises with section length from about 0.4 to 1.5 up to roughly 20,000 characters and stays between 1.2–1.6 beyond, about 2.5 mentions per ten thousand shown characters regardless of length. A deterministic recall bound makes the same point from the other side: running the rule-based extractor of the code base on the full text and on the focused text of the over-budget sections, the focused text retains 84,841 of 324,237 rule-based mentions (26.2%; 20.8% in the 10-K MD&A, 36.9% in the business item, 25.1% in the 10-Q MD&A, 84.2% in 8-K bodies), an upper bound on what any extractor can see through the windows (Section 8). A separate concern raised in review—10-K MD&As incorporated by reference and therefore too short to be submitted—turns out to be small: 169 of 29,088 parsed 10-K MD&A sections (0.6%) are under 30 words, and 430 are under 300 words with an incorporation-by-reference clause. Storing the focus spans per call is part of the proposed schema.

**Table 6.** Focus-window coverage: share of each ledgered section's text actually shown to the model, by form and item

| Form | Item | Ledgered rows | Over 9,000 chars (%) | Median chars | Share shown P10 | Share shown median | Share shown P90 | Rows shown $< 25\%$ (%) | Mentions from such rows (%) |
|---|---|---|---|---|---|---|---|---|---|
| 10-K | business | 29,285 | 81.1 | 37,168 | 5 | 16 | 100 | 66.6 | 86.1 |
| 10-K | full_body | 1,230 | 99.9 | 304,265 | 1 | 2 | 4 | 99.8 | 100.0 |
| 10-K | md_and_a | 28,919 | 94.0 | 64,194 | 4 | 9 | 24 | 90.5 | 96.7 |
| 10-K | properties | 28,236 | 5.6 | 1,255 | 100 | 100 | 100 | 3.2 | 11.3 |
| 10-K | quantitative_disclosures | 26,876 | 11.9 | 3,354 | 45 | 100 | 100 | 5.4 | 33.2 |
| 10-K | risk_factors | 29,158 | 96.8 | 62,542 | 4 | 10 | 28 | 87.9 | 93.0 |
| 10-Q | controls_and_procedures | 92,393 | 1.9 | 1,710 | 100 | 100 | 100 | 1.1 | 3.2 |
| 10-Q | financial_statements | 91,882 | 96.8 | 64,457 | 5 | 11 | 22 | 92.4 | 94.2 |
| 10-Q | full_document | 241 | 100.0 | 348,536 | 1 | 2 | 7 | 100.0 | 100.0 |
| 10-Q | legal_proceedings | 66,998 | 3.3 | 522 | 100 | 100 | 100 | 1.7 | 10.9 |
| 10-Q | md_and_a | 91,371 | 94.4 | 47,488 | 6 | 13 | 31 | 84.7 | 89.8 |
| 10-Q | quantitative_disclosures | 83,138 | 4.9 | 1,510 | 100 | 100 | 100 | 1.0 | 7.7 |
| 10-Q | risk_factors | 72,560 | 22.9 | 969 | 7 | 100 | 100 | 17.7 | 42.9 |
| 10-Q | unregistered_sales | 80,210 | 6.9 | 3,886 | 100 | 100 | 100 | 4.3 | 5.7 |
| 8-K | full_body | 400,601 | 8.8 | 3,816 | 100 | 100 | 100 | 3.3 | 5.7 |
| 10-K | all items | 143,704 | 59.3 | 28,255 | 5 | 22 | 100 | 52.1 | 84.0 |
| 10-Q | all items | 578,793 | 35.5 | 3,503 | 8 | 100 | 100 | 31.4 | 79.2 |
| 8-K | all items | 400,601 | 8.8 | 3,816 | 100 | 100 | 100 | 3.3 | 5.7 |

The snapshot's `_focus` function was replayed on all 1,123,098 ledgered sections (the replicated spans equal the real output). Sections up to 9,000 characters are shown whole; longer sections are reduced to windows around candidate-gate hits up to the budget, so the "share shown" is the fraction of the section's characters contained in the text sent to the model. The last column is the share of the item's ledger mentions that come from sections of which less than a quarter was shown (193,416 of 308,477 ledger mentions overall). No user message reached the 14,000-character cut. Shares are over all ledgered rows, including gate-rejected ones (calls-only shares in `results/focus/focus_coverage_calls_only_by_form_item.csv`). Coverage of the text is not recall: what the model missed inside the shown windows is unmeasured.

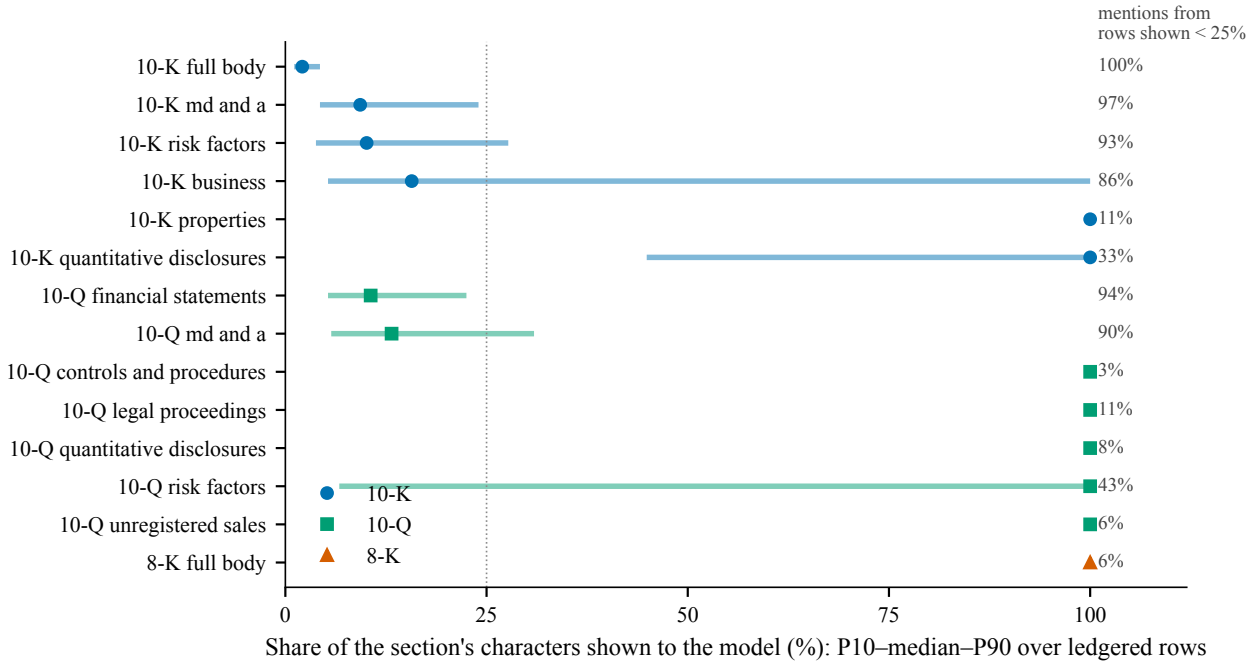



**Figure 3.** Focus-window coverage by form and item: the fraction of each ledgered section's text that reached the model. Bars span the 10th to 90th percentile over the item's ledgered rows and markers show the median (items with at least 1,000 rows); the right-hand labels give the share of the item's ledger mentions that come from sections of which less than a quarter was shown. Coverage of the text is not recall: what the model missed inside the shown windows is unmeasured (Table 6).

### 5.4. The extraction contract as implemented

The LLM call is a chat completion with a fixed system prompt, the section name and the focused text as user message, temperature 0, the JSON-object response format and a 2,000-token completion limit. The response format guarantees syntactically valid JSON; it does not enforce a schema. The field list lives in the prompt (Appendix B): the model is instructed to act as a financial analyst extracting corporate projects, is given the 21 allowed types, and is asked for a JSON object with one entry per project and the fields `project_name`, `project_type`, `description` (one sentence grounded in the text), `objective`, `status` (planned, in_progress, completed, mentioned), `amount_usd`, `locations`, `technologies`, `partners`, `benefits`, `risks`, `years` and `confidence`. Four rules in the prompt define the construct operationally: extract only real initiatives, not boilerplate or hypotheticals; return an empty list when no concrete project is described; do not invent facts—numbers and locations must appear in the text; prefer fewer high-quality projects. The output is then normalized deterministically, and several steps of that normalization were undocumented in the earlier version of this paper:

- the amount is coerced to a float in USD; `amount_raw` is *not* the string from the filing but a re-formatting of that float ("$1,500,000,000"), left empty when the amount is zero, so currency,

scale and original wording are not retained (584 mentions with a zero amount have an empty `amount_raw`; 24 carry a negative amount);

- when the model returns no `years` or no `technologies`, both are back-filled by regular expressions applied to the model's own one-sentence description rather than to the source text; the stored lists equal that regex output for 39% of mentions with years and 46% of mentions with technologies, upper bounds on the share that was back-filled;
- a missing or non-numeric `confidence` is set to 0.6, which passes the 0.45 threshold; 3,278 stored mentions carry exactly that value (304 from June, 2,974 from October) and cannot be separated from genuine model confidences of 0.6;
- an out-of-vocabulary type is mapped to `technology_deployment` and an unknown status to `mentioned` (Section 5.1);
- list fields are truncated to eight items of at most 120 characters, years are parsed as four-digit integers, and mentions with confidence below 0.45 are dropped.

The mention identifier is a SHA-1 hash of the accession number, the section identifier, the type and the first 40 characters of the project name, and mentions are written with an insert-or-replace statement. Two same-type projects whose names share their first 40 characters in one section therefore overwrite each other after being counted in the ledger, which is why the ledger sum of 308,477 mentions returned by the model exceeds the 308,065 stored mentions by 412 (10-K 127, 10-Q 257, 8-K 28) over 304 sections (Table 4). Every count in Section 7 uses the stored mentions. An index or the full name in the hash is part of the proposed schema.

What the contract does *not* guarantee should be equally explicit. The "description" is a model paraphrase, not a verbatim quote, and the character span of the supporting passage is not stored; whether the description is faithful to the passage can only be judged by a reader who opens the section. The structured fields, by contrast, are literal tokens whose presence in the stored section text is a string search, and the earlier statement that grounding was "not mechanically checkable" was wrong: Section 8 reports the containment rate of every year, partner, location, technology tag and amount for all 308,065 mentions. The model may extract a counterparty's initiative as the issuer's (the third-party label above). It may split one initiative into several entries when a section describes phases, or merge phases into one. It receives no information about the issuer other than the text, which protects against hallucinated background knowledge but also prevents it from recognizing a subsidiary name. And the same section is seen once; a project described across two sections of the same filing yields two mentions with distinct identifiers.

The instrument is re-runnable and ledgered, not deterministic. The calls used temperature 0 without a `seed` parameter, the `system_fingerprint` returned by the API was not stored, and the provider does not guarantee identical outputs across calls or model builds at temperature 0; a re-run can return different mentions. What is reproducible from the archived materials is the mapping

from the stored mentions to every table and number of this paper. The stability of the output under re-submission of the same sections, under a paraphrased prompt and under a substitute model has not been measured: a protocol over a few hundred sections is specified in Section 8 and requires new model calls that were not authorised for this revision.

### 5.5. Cost, throughput and the extraction ledger

Every ledgered section is written to the `extracted_sections` table (section, accession, backend, number of mentions, timestamp), which makes the step resumable and prevents double billing. Restated on calls rather than rows, the October 2026 run wrote 968,134 rows of which 608,439 were calls, and the June block 154,964 rows of which 95,286 were calls; the model returned at least one project for 23.9% of calls (Table 7) (15.0% of rows; 25.6% and 23.6% by run), 0.44 mentions per call (0.27 per row). At the authors' recorded total of about 775 USD at the then-current `gpt-4o-mini` prices, entered in the research log against the October run, the cost is about 0.0013 USD per call (0.0008 USD per ledger row, the figure the earlier version reported "per section"); the invoice is not archived, and if the recorded total covered the June block as well, as the review's constraints note states, the figure would be 0.0011 USD per call over the 703,725 calls of both runs—the basis is to be confirmed by the authors. The run was executed with 48 concurrent workers; the ledger timestamps span 13.7 hours of wall-clock time (2026-10-27 21:35 to 2026-10-28 11:15), of which 5.3 hours are the largest gap, so the effective throughput over 8.4 hours was 31.9 rows or 20.1 calls per second; the "twelve hours" and "27–36 sections per second" of the earlier version were mutually inconsistent and are withdrawn.

**Table 7.** Extraction ledger reconciliation by form and run: ledgered rows, gate-passing calls, productive sections and mentions

| Form | Run | Ledgered rows | Rows $<30$ words | Gate-passing calls | Productive sections | Productive (% of calls) | Mentions (ledger) | Mentions stored | Mentions per call |
|---|---|---|---|---|---|---|---|---|---|
| 10-K | June run (backfilled) | 27,164 | 0 | 26,336 | 6,204 | 23.6 | 15,787 | 15,787 | 0.60 |
| 10-K | October run | 116,540 | 0 | 108,974 | 24,673 | 22.6 | 65,835 | 65,708 | 0.60 |
| 10-Q | June run (backfilled) | 23,873 | 1,069 | 14,590 | 2,969 | 20.4 | 5,817 | 5,817 | 0.40 |
| 10-Q | October run | 554,920 | 0 | 349,383 | 72,736 | 20.8 | 146,444 | 146,187 | 0.42 |
| 8-K | June run (backfilled) | 103,927 | 1 | 54,360 | 15,234 | 28.0 | 18,326 | 18,326 | 0.34 |
| 8-K | October run | 296,674 | 0 | 150,082 | 46,414 | 30.9 | 56,268 | 56,240 | 0.37 |
| All forms | June run (backfilled) | 154,964 | 1,070 | 95,286 | 24,407 | 25.6 | 39,930 | 39,930 | 0.42 |
| All forms | October run | 968,134 | 0 | 608,439 | 143,823 | 23.6 | 268,547 | 268,135 | 0.44 |
| All forms | Both runs | 1,123,098 | 1,070 | 703,725 | 168,230 | 23.9 | 308,477 | 308,065 | 0.44 |

The run is identified by the ledger timestamp: every row of the June (S&P 500) run was re-inserted at a single instant (2026-10-27 13:12:32) and the October run occupies all later timestamps; the ledger itself stores no run, model or prompt identifier. "Gate-passing calls" are rows whose stored text matches the candidate gate of Appendix B (recomputed with DuckDB RE2 and with Python `re`; the two agree on every row and no row with mentions fails the gate); the remaining rows were ledgered as "no project" without an API call. Rows under 30 words exist only in the backfilled block, which ignored the word filter. A productive section has at least one stored mention. The ledger cannot distinguish an empty answer from a failed call (Table 8), so productive shares are lower bounds on the share of calls that would return a project under a complete run.

The ledger also records a failure mode encountered during the run, which this revision has reconciled against the run log rather than reported from memory (Table 8). When the API key's credit balance was exhausted (log window 2026-10-28 01:10 to 2026-10-28 06:22), the extractor's retry logic returned empty results that were recorded as "extracted with zero mentions": the log carries 94,465 retryable HTTP errors and 2 connection resets in that window (94,593 retryable lines in all, the remaining 128 in the twenty seconds after it), batches 47–60 (01:44–06:05) returned no mention at all, and 149,352 rows were removed from the ledger—a number that reconciles exactly with the log as the sum of sections reported complete (1,117,479) minus the sections pending when the run was restarted (968,127). The final ledger has no row inside the largest gap (01:25 to 06:40); the 504 rows stamped in the first quarter of an hour of the window survived because their calls had succeeded, and all of them are productive (1,011 mentions, 0 rows with zero mentions). Two residues remain. First, the extractor also returns an empty list after three failed attempts on any other error, and the log shows 14 JSON-decode failures outside the outage (nine before, five after; 3 in the June run), each ledgered as "no project"; the earlier version reported one. Second, the first batch after the restart (06:38–06:40; 5,198 sections in the log, 4,769 ledgered rows of 19 issuers, 2,686 calls) returned 34 mentions in the log (30 ledgered) with no logged failure: 23 productive sections against about 536 expected at the form-by-item rates of the rest of the run, with 99.2% zero-mention

calls in its last minute. The most likely reading is a block of empty answers returned with a success status while the quota was being restored; those sections are ledgered as "no project" and were not re-run. A re-submission of the 2,686 gate-passing sections of that batch (about 2,686 calls) was requested during the review and is not granted in this revision, so the batch stays flagged as suspect in Table 23 and the issuer list is published with the audit outputs. The general point stands: a ledger that cannot distinguish "no project" from "call failed" or "empty answer" is not sufficient, and an explicit `call_status` column, together with the raw response, is part of the proposed schema.

**Table 8.** Reconciliation of the October 2026 extraction outage: log, ledger and the suspect post-resume batch

| Quantity | Value |
|---|---|
| *Panel A: rows removed from the ledger after the credit-exhaustion outage (research log: 149,352)* | |
| Sections reported complete by the October log, all batches (∑ "Extraction complete") | 1,117,479 |
| Sections pending at the second start of the run | 968,127 |
| Difference = sections ledgered during the outage and re-submitted | 149,352 |
| Ledger rows with a timestamp inside the largest gap (01:25–06:40, 5.25 h) | 0 |
| Log batches returning zero mentions inside the outage window (batch ids 47–60) | 14 |
| *Panel B: failed-call lines in the October log (94,609 lines; the ledger records none of them)* | |
| Retryable HTTP 429/5xx, all lines | 94,593 |
| of which inside the outage window 01:10–06:22 | 94,465 |
| of which in the twenty seconds after it (to 06:22:21, the end of the log window) | 128 |
| Connection resets | 2 |
| JSON-decode failures outside the outage (ledgered as "no project": silent empty results) | 14 |
| "Section failed" lines | 0 |
| JSON-decode failures in the June log (of 153,894 sections) | 3 |
| *Panel C: first batch after the resume (06:38–06:40, 19 issuers)* | |
| Sections in the batch (log) / ledgered rows | 5,198 / 4,769 |
| Mentions returned (log) / ledgered | 34 / 30 |
| Gate-passing calls | 2,686 |
| Productive sections observed | 23 |
| Productive sections expected at the form × item rates of the rest of the run | 536.2 |
| Failed-call lines logged inside the batch window | 0 |

Panel A reconciles the research log's figure with the run log and the ledger timestamps; the ledger stamps every row individually (968,134 distinct timestamps), so the per-flush test proposed at review is replaced by per-minute and per-batch productivity tests (`results/ledger/`). Panel B classifies the "LLM call failed" lines of the log by reason; a JSON-decode failure returns an empty list that the ledger records as zero mentions, indistinguishable from "no project". Panel C: the first batch after the resume returned 23 productive sections where 536 were expected, with no failure logged, which is consistent with HTTP-200 empty answers while the quota was being restored; only a re-submission of its gate-passing sections (authorisation request AR-LLM-4, not granted in this revision) can separate "no project" from "empty answer". Expected counts apply the productive share of each form × item outside the batch to the batch's calls.

# 6. Resolution, Timelines and Amounts

## 6.1. The resolution rule and what it keys on

Mentions are grouped into projects by a deterministic key computed in SQL (`spid/resolve.py`): the issuer CIK, the project type, and either the lower-cased first location of the mention or, when no location was extracted, the first token of at least four letters in the normalized project name ("general" when none exists). The project identifier is the MD5 hash of that key. Within a group, the representative name and description are those of the highest-confidence mention, the canonical location is the group's location token, the technologies are the union of the mentions' technologies, the status is that of the *latest* mention by filing date, the amount is the *maximum* across mentions, and the timeline is the ordered list of mentions with their filing date, form, status, amount, confidence and snippet. First and last cited years are the minimum and maximum of the years mentioned in the text; first and last *seen* are the filing dates of the earliest and latest mention.

Rebuilding the key from the stored mentions reproduces the 121,099 project identifiers exactly and shows what the key is made of. 45.1% of projects (54,658; 38.6% of mentions) are keyed on a location, 54.8% (66,405 projects, 60.8% of mentions) on a name token, and 0.2% on the residual "general". The name token is almost always a generic head word: the ten most frequent are `acquisition` (3.4%), `share` (3.1%), `expansion` (1.8%), `stock` (1.7%), `investment` (1.6%), `merger` (1.3%), `sale` (1.3%), `development` (1.2%), `strategic` (0.9%), `capital` (0.8%) (shares of token-keyed projects), and the single token "acquisition" absorbs 40,407 mentions (21.4% of token-keyed mentions) into 2,256 projects, that is, every acquisition of an issuer whose name begins with that word is one project per issuer. The location branch is coarse as well: 8,126 projects are keyed on a country and 6,110 on a U.S. state (26% of location-keyed projects; "united states" alone keys 2,551 projects). The representative fields are not unique either: 33,820 projects have more than one mention at the maximum confidence (20,118 of them with different names, 29,954 with different descriptions) and 972 have more than one status on the latest filing date, and the SQL picks among ties in execution order. Re-running the authors' SQL reproduces every project identifier but changes the representative name of 394 projects and the description of 600; a deterministic tie-breaker (confidence, then mention identifier) is idempotent and would change 11,484 names, 503 statuses and 18,256 descriptions relative to the stored table. Until that tie-breaker is adopted, every representative name, description and status in this paper, including those of Table 16, is one admissible choice among ties rather than a reproducible value.

The rule is transparent, cheap and reversible (the mentions are untouched), and it behaves well for the common case of a recurring physical program at one site. Its failure modes are equally clear, and two of them are now measured:

- **False merges.** All mentions of one type at one location collapse into one project: every acquisition

closed by a serial acquirer in the same city, every expansion of a multi-phase site (the Cheniere example of Table 16), every drug candidate of a pharmaceutical company with no location, which fall back to a name token that is often shared ("phase", "program", "study"). The mean of 4.99 mentions per `drug_pipeline` project versus 1.63 for `technology_deployment` partly reflects this. Among `ma_integration` projects, 14.6% carry three or more distinct project names across their mentions (91.7% of those with more than ten mentions), 10.9% of those with a partner carry three or more distinct first partners, and the project-level amount of 1,630 of 12,372 M&A projects where both are named comes from a mention naming a different counterparty than the representative mention (2,347 of 21,908 across all categories); these are over-merges by construction.

- **False splits.** The same initiative named differently across filings ("Project Atlas" in 2019, "our ERP transformation" in 2021) or located at a different granularity ("Texas" vs. "Austin, Texas") yields distinct keys; so does a change of category by the model.
- **No cross-issuer identity.** A joint venture disclosed by both partners is two projects; a project inherited through an acquisition is a new project of the acquirer.

The 60.2% of projects with a single mention are the population in which false splits, true one-off disclosures and false positives are mixed, and the projects with more than ten mentions (3.1%) are where false merges concentrate. The specification's requirement—explainable, reversible merge/split decisions with match scores, resolver version and human adjudication of ambiguous clusters—is not met by the current rule and is the second schema change proposed: a `project_mention_links` table with a match score and an adjudication status, fed first by the deterministic key (score 1 by construction) and then by a probabilistic linker that uses name similarity, embedding similarity, location and time (Li et al., 2020; Peeters et al., 2023), with ambiguous clusters routed to review.

### 6.2. Sensitivity of the project count to the key

Because the key is a modelling choice, the project count is reported in this revision as a range. Table 9 and Figure 4 recompute it for nine keys and five confidence thresholds from the same 308,065 mentions. At the production threshold of 0.45, grouping by issuer and type alone gives 21,027 projects; dropping the type from the current key gives 96,749; keying on the name token only gives 114,244; the current key gives 121,099; normalizing city and state spellings gives 123,246; removing stop-words before taking the token gives 130,101; keying on two salient tokens gives 145,697; a strict key on the full normalized name and location gives 198,488; and treating every mention as its own project gives 308,065. Raising the threshold to 0.9 under the current key leaves 83,483 projects. The single-mention share moves with the key from 20.7% (issuer and type) to 75.8% (strict), and the Cheniere cluster of Table 16 has 8 mentions under the current key, 1

under the strict key, 40 under the name-only key and 176 under issuer-and-type. The abstract and Section 7 therefore quote 121,099 "under the current key" and 96,749–198,488 as the range over the alternative keys that keep a notion of initiative; none of these counts is a count of initiatives validated against human judgment.

**Table 9.** Specification curve of the project count over alternative resolution keys and confidence thresholds

| Resolution key | Projects ($\geq 0.45$) | Single-mention projects (%) | Projects with $>10$ mentions (%) | Mentions per project | Cheniere cluster size | Projects ($\geq 0.9$) |
|---|---|---|---|---|---|---|
| Issuer × type (no location, no name) | 21,027 | 20.7 | 29.8 | 14.65 | 176 | 16,562 |
| Current key without the type | 96,749 | 52.6 | 5.0 | 3.18 | 11 | 70,174 |
| Issuer × type × name token (location ignored) | 114,244 | 56.4 | 3.7 | 2.70 | 40 | 78,834 |
| Current key (location, else name token) † | 121,099 | 60.2 | 3.1 | 2.54 | 8 | 83,483 |
| Current key, city-normalised locations | 123,246 | 56.8 | 2.8 | 2.50 | 8 | 86,347 |
| Current key, extended stop-word list | 130,101 | 59.1 | 2.5 | 2.37 | 8 | 91,918 |
| Issuer × type × two name tokens | 145,697 | 63.0 | 1.7 | 2.11 | 8 | 100,490 |
| Strict: full normalised name + location | 198,488 | 75.8 | 0.5 | 1.55 | 1 | 135,411 |
| No resolution (one project per mention) | 308,065 | 100.0 | 0.0 | 1.00 | 1 | 207,150 |

Every key is applied to the same 308,065 stored mentions, rebuilt from `resolve.py` in a temporary table on the master node; † marks the key of the released `projects` table, whose identifier set is reproduced exactly. "Cheniere cluster size" is the number of mentions in the cluster that contains the highest-confidence mention of the Table 16 example. Thresholds drop mentions below the stated confidence before resolution. The project count is therefore a property of the key: 21,027–308,065 at the release threshold, and the single-mention share and projects-per-issuer statistics move with it. Under the current key 66,405 projects (54.8%) are keyed on a name token because the mention has no location, 54,658 on a location and 229 on the fallback "general" (as in the text; 286 keys end in the token "general" because a project name whose first token is the word "general" is not distinguished from the fallback); the single token "acquisition" absorbs 40,407 mentions into 2,256 projects (Table 31). The stored table is not exactly reproducible: 33,820 projects have several mentions at the maximum confidence (`arg_max` ties), and re-running the authors' SQL changes 394 representative names and 600 descriptions while reproducing every project identifier.

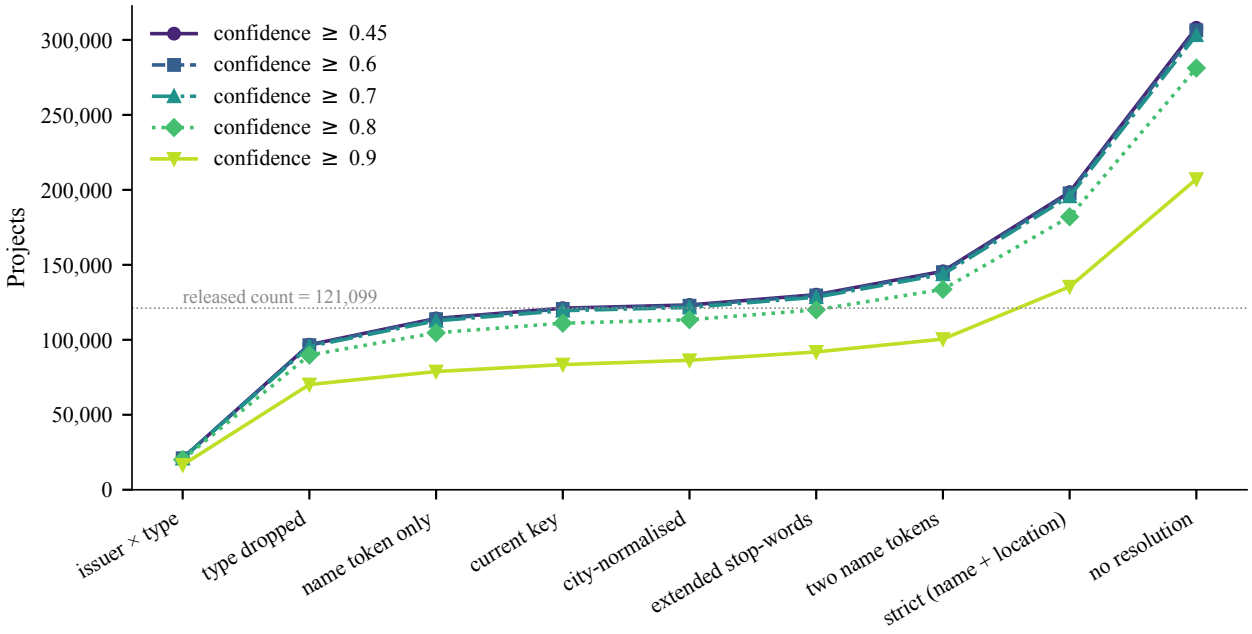



**Figure 4.** Specification curve of the project count. Each line applies one confidence threshold to the 308,065 stored mentions and resolves them under nine keys ordered by the count they produce at the release threshold; the released table corresponds to the current key at 0.45 (dotted line). The count spans an order of magnitude across keys and moves little with the threshold below 0.9, so it is a property of the resolution rule rather than a count of initiatives (Table 9). Counts are exact.

### 6.3. Timelines and point-in-time structure

Each mention is dated by the filing date of its document, and each project by the filing dates of its first and last mentions and by the years cited in the text. This is a point-in-time structure in one specific sense: the state of a project *as known at* a filing date can be reconstructed by filtering the timeline to mentions filed on or before that date, and nothing in a later filing is used to alter an earlier mention. It is *not* yet the event structure the specification describes. The database does not separate first announcement, SEC acceptance timestamp (only the filing date is stored), referenced period, expected start, observed start, expected completion, realized completion and cancellation; cited years are stored as an unordered list without roles, and for up to 39% of mentions they may have been read from the model's paraphrase rather than from the text (Section 5.4); and a project's status is the latest statement rather than a dated transition, with ties on the latest date resolved arbitrarily for 972 projects. The proposed `project_events` table (Appendix C) would record each such event with its date, its evidence mention and a flag distinguishing information available at the acceptance date from information revised later.

### 6.4. Amounts

The amount of a mention is whatever USD figure the model associated with the project in the passage, coerced to a float; the source wording is not retained (Section 5.4), so currency, scale and basis cannot be recovered from the stored data. The project-level `total_amount_usd` is the *maximum* over mentions, chosen in the June 2026 release because the same budget is typically restated in successive filings and summing would multiply it by the number of mentions. The maximum avoids that error but introduces others: it cannot tell a transaction price from a capex budget, a contract value from a financing facility, an incurred cost from an estimate, or a figure for the whole program from a figure for one phase, and it is silently inflated when a merged project combines distinct initiatives (Section 6.1). The tail of the distribution shows what unchecked coercion produces: the largest stored amount is $1.0 \times 10^{12}$ USD (one mention, found as a number in a 10-K MD&A with no currency token nearby), 161 mentions in 104 projects carry amounts of 50 B USD or more, 24 amounts are negative and 584 are zero. Of the 200 largest amounts, 191 are found as a number in the source section and 178 with a currency token within forty characters, so most of the tail is transcribed rather than invented, but scale errors from tables "in thousands" or "in millions" cannot be excluded without typing; a non-USD currency token sits within forty characters of 1,837 of the 112,692 located amounts (1.6%). We therefore insist on two reading rules. First, `total_amount_usd` is "the largest amount an issuer cited in connection with this project", nothing more. Second, amounts must never be summed across projects or interpreted as realized investment. The proposed `project_amounts` table keeps every amount mention with its source string, currency, scale, reporting date, an `amount_type` (announced budget, estimated cost, capex incurred, transaction value, contract value, financing, other) and an estimate/realization flag; filling it requires either a second extraction pass or annotation, neither of which was authorised for this revision, and the release notes recommend excluding non-positive amounts and flagging amounts of 50 B USD or more until then.

### 6.5. Data model: current and proposed

Table 22 in Appendix C lists the tables of the current build alongside the tables required by the specification and marks each as implemented, partial or proposed. In summary, the current `companies`, `filings`, `filing_sections`, `spid_sections`, `project_mentions`, `projects`, `project_timeline` and `extracted_sections` tables cover universe, filings, sections, mentions, projects, a timeline and an extraction ledger; the `issuers` history, `project_mention_links`, `project_events`, `project_amounts`, `extraction_runs` (with model snapshot, seed, fingerprint, prompt and taxonomy hashes, call status and raw response) and `human_labels` tables are proposed and would be added by migration, never by destructive replacement.

# 7. Descriptive Patterns

All tables, figures and inline numbers in this section are produced by `code/export_stats.py` (read-only queries on the merged databases at revision `2324b58`, run on the master node) and `code/make_tables.py` (exports and audit outputs to exhibits and macros); the generators contain no data literal, which `tests/test_tables.py` checks, and the source and formula of every inline macro are listed in `results/inline_numbers.json`. The numbers describe *disclosed* initiatives as detected by the current pipeline and resolved by the current rule. They are not estimates of real-world project counts, they have not been corrected for the extraction and linkage errors whose measurement is described in Section 8, and the project counts hold under the current key only (Section 6.2).

## 7.1. From sections to projects

The extraction ledger contains 1,123,098 rows, of which 703,725 passed the lexical gate and were submitted to the model (Section 5.3). The model returned at least one project for 168,230 of them (23.9% of calls, 15.0% of rows), yielding 308,477 mentions before identifier de-duplication and 308,065 stored mentions—0.44 per call and 1.83 per productive section. The density differs sharply by form and by item (Table 10): the MD&A items of the 10-K and 10-Q produce 39,099 and 99,305 mentions respectively, Item 1 (Business) of the 10-K 30,492, and the 10-Q financial-statement notes 30,159, whereas risk-factor and market-risk items produce almost none despite their length—a pattern that Section 5.3 shows to be partly a property of the focus windows rather than of the text. The 8-K body yields 74,566 stored mentions (74,594 returned by the model) from 400,601 filings. Resolution reduces 308,065 mentions to 121,099 projects (2.54 mentions per project); 60.2% of projects rest on a single mention, 17.0% on two, and 8.2% on more than five (Table 13).

**Table 10.** Ledgered rows, mentions, gate-passing calls, text shown to the model and financing-class share, by form and item

| Form | Section (item) | Ledgered rows | Mentions | Gate-passing calls | Median share of text shown (%) | Financing-class mentions (%) |
|---|---|---|---|---|---|---|
| 10-K | `business` | 29,285 | 30,492 | 29,118 | 16 | 0.4 |
| 10-K | `risk_factors` | 29,158 | 2,797 | 28,941 | 10 | 0.5 |
| 10-K | `md_and_a` | 28,919 | 39,099 | 28,050 | 9 | 1.3 |
| 10-K | `properties` | 28,236 | 7,879 | 25,940 | 100 | 0.1 |
| 10-K | `quantitative_disclosures` | 26,876 | 531 | 22,032 | 100 | 7.0 |
| 10-K | `full_body` | 1,230 | 824 | 1,229 | 2 | 0.2 |
| 10-Q | `controls_and_procedures` | 92,393 | 7,082 | 20,913 | 100 | 0.2 |
| 10-Q | `financial_statements` | 91,882 | 30,159 | 90,106 | 11 | 2.4 |
| 10-Q | `md_and_a` | 91,371 | 99,305 | 90,269 | 13 | 1.4 |
| 10-Q | `quantitative_disclosures` | 83,138 | 808 | 46,720 | 100 | 9.5 |
| 10-Q | `unregistered_sales` | 80,210 | 9,535 | 61,867 | 100 | 54.5 |
| 10-Q | `risk_factors` | 72,560 | 4,420 | 33,826 | 100 | 1.0 |
| 10-Q | `legal_proceedings` | 66,998 | 845 | 20,031 | 100 | 1.0 |
| 10-Q | `full_document` | 241 | 107 | 241 | 2 | 0.0 |
| 8-K | `full_body` | 400,601 | 74,594 | 204,442 | 100 | 19.2 |

Mentions are ledger counts (before identifier de-duplication, Table 4). The 10-K `full_body` rows are filings of the June (S&P 500) ingest in which its segmenter found no Item heading; the October parser's `full_document` fallback produced 10-K rows that are excluded from extraction by scope (Section 4), while its 10-Q `full_document` rows are ledgered. "Median share of text shown" is the median, over the item's ledgered rows, of the fraction of the section's characters contained in the focus windows actually sent to the model (Table 6). "Financing-class mentions" is the share of the item's stored mentions whose name or description matches the repurchase / dividend / credit-facility / notes-offering pattern of Table 5; rows above 10% are the 10-Q `unregistered_sales` item (Part II Item 2, where issuer repurchases are tabled) and the 8-K body.

### 7.2. Composition by category, sector and status

Table 11 and Figure 5 show the distribution over the 21 categories. M&A and post-merger integration dominate (34.5% of projects), followed by capital-expenditure programs (8.9%), strategic partnerships (8.2%), geographic expansion (6.5%) and plant or site construction (6.3%). These shares include the financing-instrument class: excluding projects whose representative name or description matches the financing pattern, `capex_program` falls from 10,720 to 4,725 projects (8.9% to 4.3%), `cost_reduction` from 6,481 to 5,403 and `ma_integration` from 41,742 to 37,527 (`results/fp_classes/table5_restated_excl_financing.csv`); 12,027 projects (9.9%) match on their representative fields and 13,139 have at least one matching mention. The

technology-related categories that motivated the first SPID release—data centers, AI initiatives, cloud migration, automation—are small in count (between 264–449 projects each). Their disclosed amounts are reported in Table 11 with the number of projects that carry one and a bootstrap interval for the median: the data-center median of 166.0 M USD rests on 129 projects with an amount and the AI median of 127.5 M on 20, so the latter should not be quoted without its $N$. Two readings of the composition are possible and the data do not discriminate between them: either acquisitions are genuinely the most frequently disclosed discrete initiative in SEC filings, which is plausible given that every material acquisition triggers an 8-K and recurring 10-Q notes; or the extraction contract, which asks for "concrete initiatives", is more easily satisfied by acquisitions (named counterparty, date, price) than by internal programs. The 8-K item-code consistency check of Section 8 shows that the `ma_integration` label co-occurs with the acquisition item at a rate far above its base rate, which supports the first reading for 8-Ks without settling the second. Disentangling the two is a question for the human-annotation study.

**Table 11.** Resolved projects by taxonomy category: counts, financing-class restatement, disclosed amounts and over-merge indicator

| Category | Projects | Share (%) | Excl. financing class ($N$) | With amount ($N$) | Median amount (M USD) | 95% CI of the median | Mentions per project | $\geq 3$ distinct names (%) |
|---|---|---|---|---|---|---|---|---|
| M&A / integration | 41,742 | 34.5 | 36,687 | 28,044 | 155.0 | [150.0, 161.5] | 3.12 | 14.6 |
| Capital expenditure program | 10,720 | 8.9 | 4,519 | 9,452 | 300.0 | [300.0, 323.2] | 1.79 | 6.4 |
| Strategic partnership | 9,909 | 8.2 | 9,548 | 2,943 | 75.0 | [60.0, 81.9] | 1.77 | 6.7 |
| Geographic expansion | 7,868 | 6.5 | 7,850 | 1,431 | 36.7 | [30.1, 44.6] | 2.25 | 10.2 |
| Plant / site construction | 7,590 | 6.3 | 7,557 | 2,511 | 56.0 | [50.0, 64.9] | 2.42 | 13.5 |
| Cost reduction / restructuring | 6,481 | 5.4 | 5,397 | 4,378 | 50.0 | [45.0, 50.0] | 2.03 | 5.7 |
| Drug / clinical pipeline | 5,360 | 4.4 | 5,355 | 778 | 46.1 | [35.5, 60.0] | 4.99 | 27.9 |
| New product / launch | 5,356 | 4.4 | 5,353 | 447 | 26.7 | [22.0, 38.1] | 2.33 | 12.2 |
| R&D program | 5,080 | 4.2 | 5,073 | 1,272 | 20.0 | [18.4, 25.0] | 2.31 | 11.4 |
| Infrastructure investment | 4,481 | 3.7 | 4,452 | 1,694 | 48.0 | [41.1, 55.0] | 1.90 | 7.5 |
| Manufacturing / capacity expansion | 4,140 | 3.4 | 4,130 | 1,040 | 45.0 | [38.0, 53.2] | 2.13 | 11.3 |
| Technology deployment | 4,128 | 3.4 | 3,824 | 1,219 | 135.0 | [120.0, 150.0] | 1.63 | 4.8 |
| Digital transformation | 2,642 | 2.2 | 2,642 | 240 | 34.2 | [21.8, 50.0] | 1.99 | 6.2 |
| ERP / systems implementation | 1,735 | 1.4 | 1,735 | 184 | 15.0 | [9.3, 21.0] | 2.44 | 3.5 |
| Sustainability / ESG | 1,305 | 1.1 | 1,284 | 410 | 34.4 | [20.1, 45.0] | 2.00 | 6.1 |
| Supply-chain initiative | 684 | 0.6 | 684 | 139 | 38.0 | [25.0, 75.0] | 1.71 | 3.5 |
| Data center | 449 | 0.4 | 443 | 129 | 166.0 | [93.0, 375.0] | 1.98 | 9.3 |
| Energy transition / renewables | 415 | 0.3 | 413 | 109 | 109.0 | [85.0, 254.0] | 2.09 | 10.1 |
| AI / ML initiative | 376 | 0.3 | 376 | 20 | 127.5 | [35.0, 612.5] | 1.68 | 4.8 |
| Automation / robotics | 374 | 0.3 | 374 | 25 | 13.7 | [3.4, 38.4] | 1.68 | 2.7 |
| Cloud migration | 264 | 0.2 | 264 | 35 | 50.0 | [9.4, 250.0] | 1.97 | 5.7 |
| Total | 121,099 | 100.1 | 107,960 | 56,500 | 128.5 | – | 2.54 | |

Category codes are listed in Table 21. Shares are rounded to one decimal and sum to 100.1. "Excl. financing class" removes projects with at least one mention matching the repurchase / dividend / credit-facility / notes-offering pattern of Table 5 (13,139 projects, 10.8%); the two columns bracket the construct violation, they do not measure precision. "With amount" counts projects for which at least one mention carries a USD figure; "median amount" is the median across those projects of the largest amount cited (Section 6.4 on why amounts are never summed). Medians printed in grey rest on fewer than 50 projects with an amount (`ai_initiative`, `automation`, `cloud_migration`) and should not be quoted without their $N$. The 95% interval of the median is a percentile bootstrap over projects (B = 1,000, seed 20261105, `data/projects_by_type_ci.csv`); an interval equal to the point estimate reflects the rounding of disclosed amounts to round numbers. "$\geq 3$ distinct names" is the share of the category's projects whose mentions carry three or more distinct project names, an over-merge indicator of the resolution key (Table 9). All counts are under the current resolution key at the 0.45 confidence threshold.

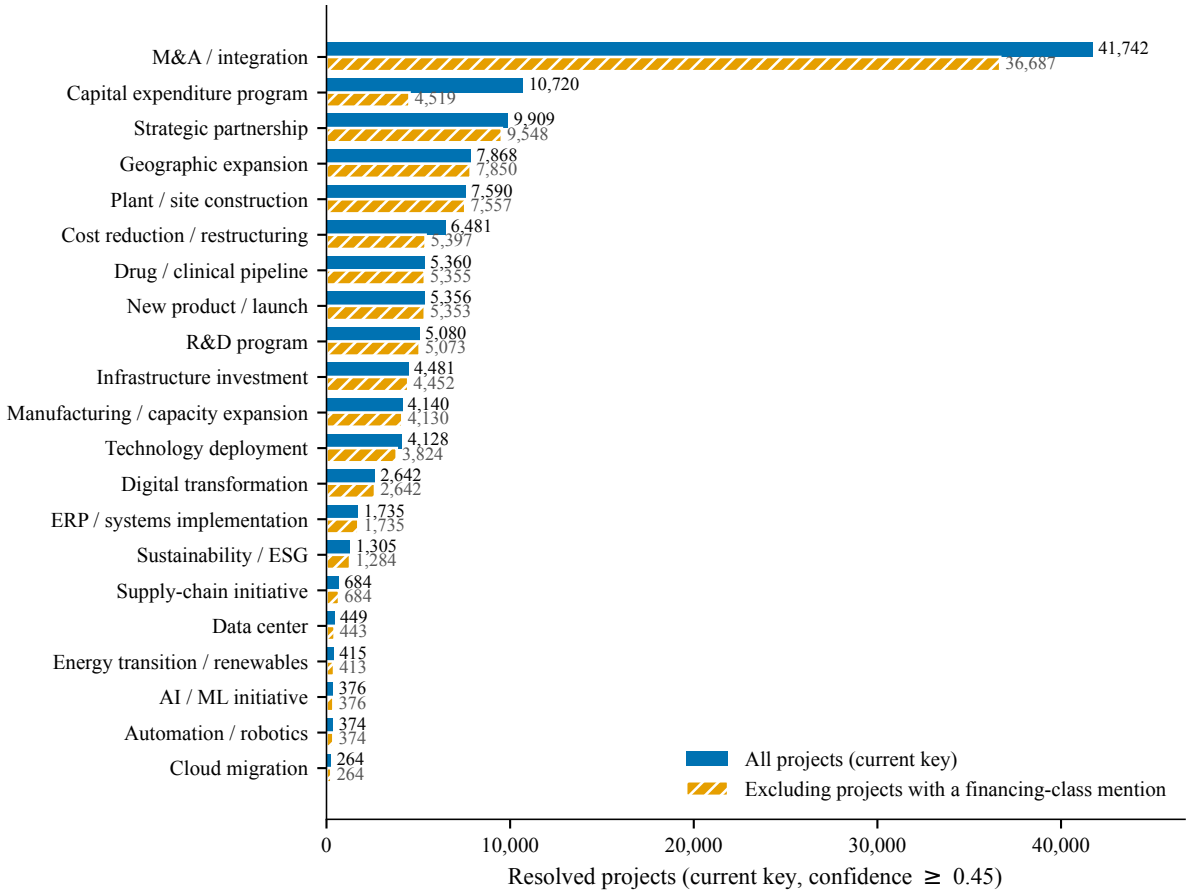



**Figure 5.** Resolved projects by taxonomy category (121,099 projects under the current resolution key). Hatched bars remove the projects with at least one mention matching the financing-instrument / buyback / dividend pattern of Table 5; the two bars bracket a construct violation detectable by pattern, not precision. Counts are exact database counts; the dependence of every count on the key is shown in Table 9.

By sector (Table 12), health care (23,679 projects, 429 issuers) and industrials (17,890) contribute the most projects; utilities, materials, energy, real estate have the most projects per issuer (62–75), and financials the fewest (32), consistent with the intuition that asset-heavy, regulated or resource-extracting firms narrate discrete physical programs while banks describe fewer discrete initiatives—and also consistent with a key that creates one project per site. The list of the most "productive" issuers (Table 28 in the appendix)—uranium and gold miners, airlines, a bitcoin miner, an oilfield-services company—is consistent with issuers whose business *consists* of projects (mines, routes, wells, data-center sites) enumerating them one by one; whether it also reflects over-splitting by the key has not been measured.

**Table 12.** Issuers and resolved projects by sector

| GICS sector (iShares) | Issuers in universe | of which S&P 500 | Issuers with projects | Projects | Projects per issuer |
|---|---|---|---|---|---|
| Health Care | 433 | 59 | 429 | 23,679 | 55.2 |
| Industrials | 393 | 79 | 386 | 17,890 | 46.3 |
| Consumer Discretionary | 276 | 48 | 267 | 15,072 | 56.4 |
| Financials | 462 | 76 | 446 | 14,223 | 31.9 |
| Information Technology | 333 | 73 | 325 | 11,756 | 36.2 |
| Real Estate | 146 | 31 | 145 | 9,001 | 62.1 |
| Materials | 126 | 26 | 117 | 7,407 | 63.3 |
| Energy | 119 | 21 | 107 | 6,710 | 62.7 |
| Consumer Staples | 103 | 36 | 103 | 5,921 | 57.5 |
| Utilities | 68 | 31 | 66 | 4,920 | 74.5 |
| Communication Services | 105 | 20 | 101 | 4,520 | 44.8 |
| Total | 2,564 | 500 | 2,492 | 121,099 | 48.6 |

Sector labels come from the iShares holdings file (Russell proxy) or, for S&P 500-only issuers, from the Wikipedia GICS table; "Issuers in universe" includes issuers without filings. Projects per issuer is projects divided by issuers with at least one project. The S&P 500 issuers were extracted in two runs and with two 10-K segmenters (Table 15), so cross-sector differences in projects per issuer mix disclosure behaviour with run and volume effects.

The latest status recorded for a project is `in_progress` for 41.6%, `completed` for 41.0%, `planned` for 15.1% and `mentioned` for 2.3% (Table 13). Because status is taken from the most recent mention, it is a snapshot of the last thing the issuer said, not the terminal fate of the project: a project that was announced, mentioned once as in progress and then silently abandoned is recorded as in progress. The span between the earliest and latest year cited *in the text* is unknown for 14.5% of projects (17,599); among the 103,500 projects with cited years it is zero for 64.9% and exceeds five years for 10.0% (55.5% and 8.5% of all projects). The specification's distinction between announcement, filing, expected start, observed start, expected and realized completion, and cancellation is not yet implemented as separate dated events (Section 6).

**Table 13.** Latest status, number of mentions and span of cited years of resolved projects

| Latest status | Projects | Mentions per project | Projects | Span of cited years | Projects |
|---|---|---|---|---|---|
| `in_progress` | 50,327 | 1 | 72,932 | 0 years | 67,170 |
| `completed` | 49,687 | 2 | 20,544 | 1-2 years | 17,307 |
| `planned` | 18,340 | 3-5 | 17,681 | 3-5 years | 8,697 |
| `mentioned` | 2,745 | 6-10 | 6,173 | >5 years | 10,326 |
| | | >10 | 3,769 | unknown years | 17,599 |
| Total | 121,099 | | 121,099 | | 121,099 |

Status is that of the project's latest mention by filing date, a snapshot of the last statement rather than the terminal fate (for 972 projects several mentions share the last filing date with different statuses, and the stored choice is one admissible value among ties). The span is the difference between the latest and earliest four-digit year cited in the mentions' text; it is unknown for 17,599 projects (14.5%), so the shares quoted in the text, 64.9% at zero span and 10.0% above five years, are over the 103,500 projects with known years. Single-mention projects (60.2%) mix false splits, one-off disclosures and false positives; projects above ten mentions are where false merges concentrate (Table 9).

### 7.3. Amounts, confidence and time

A USD amount is attached to 120,240 mentions (39.0%) and to 56,500 projects (46.7%). Table 14 reports the distribution of the largest amount cited per project: median 128.5 M USD, interquartile range 23–500 M, and a long right tail (P99 of 12.5 B USD; maximum $1.0 \times 10^{12}$ USD; 104 projects at or above 50 B USD; Section 6.4 on the grounding of the tail). The sum of these maxima across projects is 55.9 trillion USD, a number we report only to make the point that it must *not* be interpreted: it adds transaction values, capex budgets, contract values, financing amounts (17.1% of mention amounts match the financing pattern) and estimated costs across projects and years, it double counts initiatives that the resolver split, and it treats a stated budget as if it were spent. Section 6.4 explains how amounts are stored and what an `amount_type` field would need to add.

**Table 14.** Distribution of the largest disclosed amount per project and of projects per issuer

| | P10 | P25 | Median | P75 | P90 | P99 |
|---|---|---|---|---|---|---|
| Largest amount cited per project, M USD ($N$ = 56,500) | 3.2 | 23 | 128.5 | 500 | 1,503 | 12,501 |
| Projects per issuer ($N$ = 2,492) | 12 | 24 | 42 | 67 | 94 | 140 |

Amounts are the largest USD figure cited across the mentions of a project; they mix transaction values, budgets, contract values, financing facilities and estimates, and are never summed in the text. The maximum is 1,000 B USD (one mention, number found in the section text, no currency token nearby); 161 mentions in 104 projects cite 50 B USD or more, of which 155 have the number in the text and 144 a currency token within 40 characters; 584 mentions carry an amount of exactly zero (with `amount_raw` NULL) and 24 a negative amount. 259 projects have a non-positive maximum amount; none of these values has been corrected (author decision pending). Projects per issuer are counted under the current resolution key, whose choice moves the total by a factor of ten (Table 9). Quantiles are exact order statistics of the stored tables; no sampling uncertainty applies.

The self-reported confidence of mentions is concentrated at the top of the scale (Figure 6; Table 29 in the appendix): 35.2% of mentions have confidence 1.0, 32.7% have 0.9, and only 1.6% fall at or below 0.6. Mentions below 0.45 are discarded at extraction time, and the 0.6 bin contains the 3,278 mentions whose confidence the post-processing defaulted because the model returned none (Section 5.4). The distribution says more about the model's reporting habits than about accuracy; the only accuracy-related quantity available so far moves the wrong way with confidence—the share of mentions whose structured fields are all found in the source text is 86.4% in the 0.9–1.0 bin against 90.7–94.2% in the lower bins (Section 8)—and whether a confidence of 0.7 corresponds to a lower precision than a confidence of 1.0 is precisely the calibration question that only labeled data can answer.

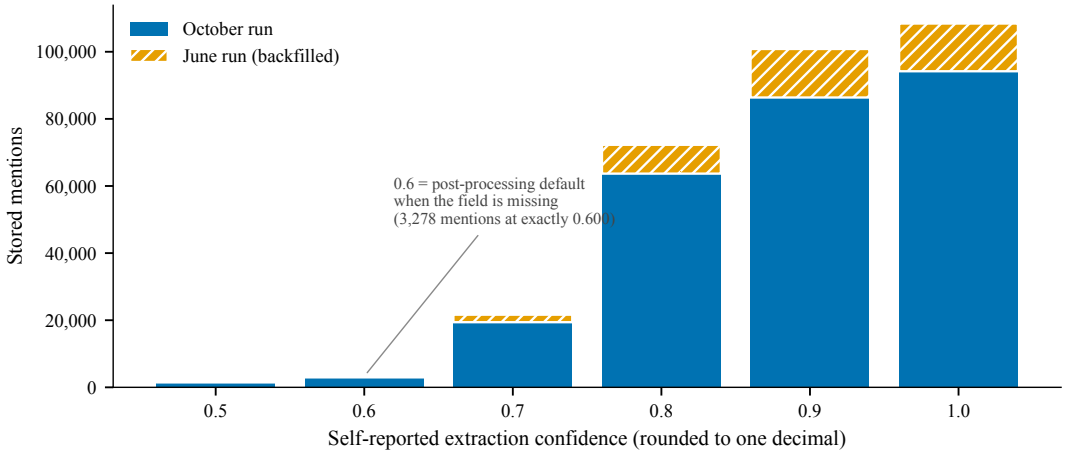



**Figure 6.** Self-reported extraction confidence of the 308,065 stored mentions, by run (mentions below 0.45 are discarded before storage). The value is the model's own JSON field, rounded to one decimal; a missing field is set to 0.6 by the post-processing code. The distribution describes a reporting habit, not accuracy: calibration requires the human labels of gate (d). Counts are exact.

Over time (Figure 7; Table 26), mentions rise from 11,908 in 2010 to 23,220 in 2025, almost in proportion to the number of issuers observed (1,224 to 2,280 with at least one mention; 9.7 to 10.2 mentions per such issuer), while the number of projects first observed in a year lies between 6,240–8,216 over 2011–2025. The first and last years are excluded from that range on purpose: 2010 is left-censored, since every initiative already under way when the corpus starts is "first seen" then (7,835 projects), and 2026 is partial (6,694 projects to October 27). The rise in mentions is consistent with a coverage effect of the snapshot universe rather than with an increase in disclosed activity: holding the panel fixed at the 1,389 issuers that file in every year from 2010 to 2025, mentions per issuer move from 9.6 to 8.8 and newly seen projects per issuer from 4.7 to 2.8 (6,533 to 3,926 projects; `results/timeseries/first_seen_fixed_cohort.csv`). The decline in new projects per issuer is, at least in part, mechanical: under a time-invariant key, later mentions of a serial acquirer or of a multi-phase site attach to an existing project rather than opening a new one. A windowed variant that opens a new episode when a key is silent for more than three (five) years counts 131,823 (126,318) episodes instead of 121,099 projects, and its first-seen series is reported alongside the current one in Table 26. Both readings remain testable with the panel; neither has been tested against labels.

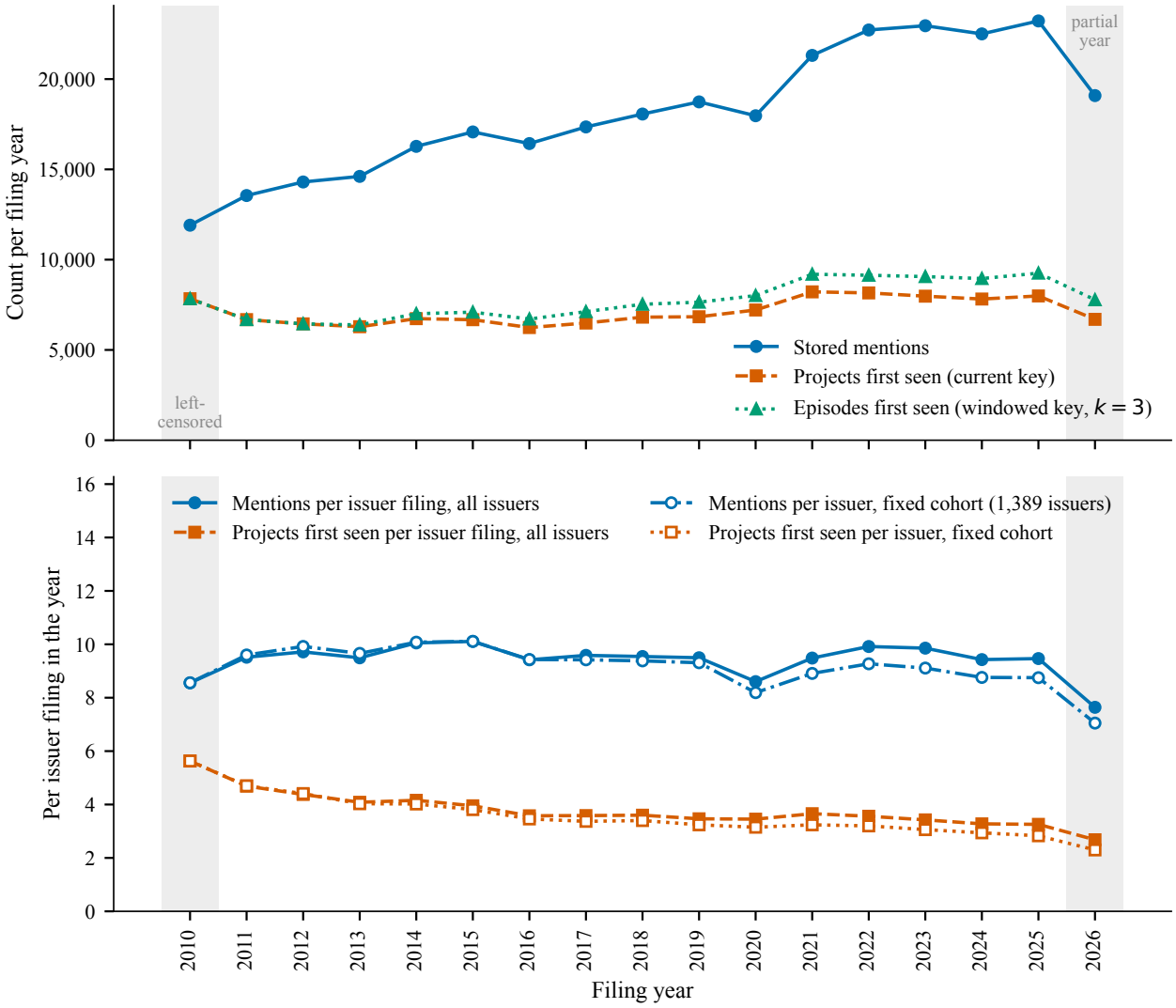



**Figure 7.** Mentions and projects first seen by filing year. Top: stored mentions, projects first seen under the current key and episodes first seen under a windowed key that splits a project when consecutive mentions are more than three years apart. Bottom: the same quantities per issuer filing in the year, for all issuers (filled markers) and for the fixed cohort of issuers filing in every year 2011–2025 (open markers), which holds the panel composition constant. 2010 is left-censored and 2026 is partial (shaded); both axes start at zero. The flat mentions-per-issuer series and the falling new-projects-per-issuer series of the cohort are consistent with a time-invariant key attaching later mentions to existing projects; the figure does not identify disclosure behaviour. Counts are exact.

The two extraction runs add a confound of their own to any comparison between the S&P 500 cohort and the rest of the universe. Issuers in both lists carry 55.8 projects per issuer against 46.9 for Russell-only issuers, which the earlier version read as a difference in disclosure behaviour; counting only projects touched by an October-run mention, the same issuers carry 25.1. In an issuer-level regression of the project count on an S&P 500 indicator (2,499 issuers, HC1 standard errors), the coefficient is 8.84 (1.62) with all mentions, −22.98 (1.20) with October mentions

only, and −0.28 (1.52) once the logarithm of the issuer's ledgered sections is controlled for (`results/runs/table6_cohort_regression.json`). The cohort difference is a difference in run and volume, not evidence about behaviour, and Appendix E gives the reconciliation by run and form (Table 15).

**Table 15.** Projects per issuer by index-membership cohort: the apparent S&P 500 effect is a run and volume effect

*Panel A: projects per issuer by cohort and by run of the mentions counted*

| Cohort | Issuers | Projects (all mentions) | Per issuer | Projects (October mentions) | Per issuer | Ledger rows per issuer | October rows per issuer |
|---|---|---|---|---|---|---|---|
| S&P 500 ∩ Russell proxy | 491 | 27,382 | 55.8 | 11,738 | 25.1 | 552.4 | 242.7 |
| Russell proxy only | 1,992 | 93,333 | 46.9 | 93,333 | 46.9 | 423.8 | 423.8 |
| S&P 500 only | 9 | 384 | 42.7 | 118 | 13.1 | 521.0 | 201.4 |

*Panel B: issuer-level OLS of projects on the S&P 500 cohort dummy (HC1 standard errors in parentheses)*

| Dependent variable | S&P 500 dummy | | log(1 + ledger rows) | | $N$ | $R^2$ |
|---|---|---|---|---|---|---|
| Projects, all mentions | 8.84 | (1.62) | – | | 2,499 | 0.012 |
| Projects, October mentions only | -22.98 | (1.20) | – | | 2,499 | 0.083 |
| Projects, all mentions, volume control | -0.28 | (1.52) | 22.09 | (0.73) | 2,499 | 0.274 |
| Projects, October mentions, volume control | -10.59 | (1.09) | 17.54 | (0.51) | 2,499 | 0.323 |

Panel A counts projects per issuer with all mentions and with only the mentions produced by the October run (a project is "October" when at least one of its mentions carries an October timestamp); S&P 500 issuers were additionally extracted in the June run, whose 10-K items came from a different segmenter (Section 4). Panel B reports the coefficient on the S&P 500 dummy from issuer-level OLS with heteroskedasticity-robust (HC1) standard errors; the sign flips from +8.8 to −23.0 projects per issuer when the run is held fixed and vanishes once the number of ledgered rows is controlled for. Economic magnitude: one log point of ledgered rows is associated with 22.1 additional projects. No causal reading is intended: the regressions describe the composition of the database, not disclosure behaviour.

### 7.4. A provenance example

Table 16 follows one resolved project through its mentions: the Corpus Christi liquefaction trains of Cheniere Energy (ticker LNG), typed `plant_construction`, with 8 mentions in 8-K filings between 2014-04-08 and 2022-03-07 and a largest cited amount of 11.5 B USD. Each row is traceable: the accession number identifies the document on EDGAR, the section identifier locates the passage in the parsed corpus, and the snippet is the model's one-sentence description, which the contract requires to be grounded in the passage but which is not itself checked (Section 5.4). The example also exposes the resolver's coarseness. The project key for this issuer and type is the location string "san patricio county, texas", so the Stage 3 expansion and the earlier trains share one project identifier although Cheniere describes them as distinct phases; the representative name, "Corpus Christi Liquefaction Stage 3 Project", is that of the highest-confidence mention, filed in 2022, and is applied retroactively to mentions from 2014 that describe the first trains; and that

representative is one admissible choice among tied mentions (Section 6.1). Under the strict key the same mentions form 1 project each; under the issuer-and-type key they join 176 mentions. A reviewer can see this because the mentions and their snippets are preserved; a user who consumes only the `projects` table cannot.

**Table 16.** Provenance of one resolved project: Cheniere Energy, Corpus Christi liquefaction (project `3e2705ba...`)

| Accession | Filed | Form | Status / amount | Evidence snippet (model-generated description) |
|---|---|---|---|---|
| 0000003570-14-000070 | 2014-04-08 | 8-K | planned / – | The construction of the first liquefaction train at the Corpus Christi Liquefaction facility is contingent upon receiving regulatory approvals and securing financing. |
| 0000003570-14-000114 | 2014-05-30 | 8-K | planned / – | CCLNG will construct a second liquefaction train at its facility to sell LNG to IBERDROLA under a long-term agreement. |
| 0000003570-14-000117 | 2014-06-02 | 8-K | planned / – | Cheniere Energy, Inc. plans to construct a second liquefaction train at the Corpus Christi Liquefaction facility. |
| 0000003570-14-000139 | 2014-06-30 | 8-K | planned / – | Cheniere Energy, Inc. plans to construct a second liquefaction train at the Corpus Christi Liquefaction facility. |
| 0000003570-14-000153 | 2014-07-17 | 8-K | planned / – | Cheniere Energy, Inc. plans to construct a third liquefaction train at its Corpus Christi Liquefaction facility. |
| 0000003570-14-000306 | 2014-12-18 | 8-K | planned / – | Cheniere Energy, Inc. plans to construct a third liquefaction train at its Corpus Christi facility. |
| 0001193125-15-186045 | 2015-05-13 | 8-K | in_progress / 11.5 B USD | The project involves the development, construction, operation, and maintenance of a natural gas liquefaction and export facility and related infrastructure in San Patrici |
| 0000003570-22-000029 | 2022-03-07 | 8-K | in_progress / 5.5 B USD | The project involves the engineering, procurement, construction, pre-commissioning, commissioning, start-up, and testing of a mid-scale natural gas liquefaction facility |

Snippets are the first 170 characters of the model-generated description stored in `project_timeline.snippet`; the description is a paraphrase, not a quotation, and the character span of the supporting passage is not stored; the full source section is retrievable through `project_mentions.section_id`. All 8 printed mentions are 8-K filings. The representative name is that of the highest-confidence mention (filed 2022) applied retroactively to the whole timeline, and is one admissible choice among mentions tied at the maximum confidence. The key for this issuer and type is the location "san patricio county, texas" (`results/resolver/example_provenance_keys.csv`), so distinct phases at that site share one identifier; the cluster that contains the example's highest-confidence mention holds 8 mentions under this key, 1 under the strict name $\times$ location key and 176 under the issuer $\times$ type key (Table 9).

### 7.5. Knowledge graph and research-ready exports

The resolved projects are projected into a graph of 183,472 nodes (2,492 companies, 121,099 projects, 14,508 locations, 8,551 technologies, 36,822 partners) and 204,456 edges (`owns`, `located_in`, `uses`; Table 30 in the appendix); partner nodes are currently created without edges, a known gap, and `uses` edges inherit the technology tags of which up to 46% may have been back-filled from the description (Section 5.4). Each project also has a 384-dimensional sentence embedding of its name, type, description and technologies, which supports semantic search through the companion web application. Alongside the project layer, the corpus layer publishes the firm-year and firm-quarter panels of thematic factor scores (32,852 and 128,043 rows for 2,499 issuers with filings) used in the sibling UQO working paper on AI disclosure (Kammoun et al., 2026), now covering the full universe. A firm-year aggregation of the *mentions*, by contrast, is not yet available as stated in the earlier version: `fiscal_year` is empty for 68,827 mentions (all 39,930 June mentions and 28,074 October 10-Q mentions among others) because the corresponding filings carry no fiscal year, and a derived fill from the filing date is exported with the audit outputs but not stored (Appendix E).

## 8. Validation Framework and Current Audit

The aim of identifying "all projects" of an issuer is a *coverage target*, not an assumption. Only projects that appear in the captured sources, survive parsing and gating, fall inside the focus windows, are recognized by the model and are correctly linked are observable, and each of these steps loses or distorts some. In the framework of Ludwig et al. (2026), the output of Section 5 is a model-generated measurement whose error structure is unknown until a validation subsample with independent labels identifies it; until then it can describe the corpus but cannot serve as a regressor without an explicit measurement-error model. This section specifies how the losses are to be measured and reports what has been measured. The headline is in three parts: the *coverage* and *integrity* components are implemented and quantified (Section 8.1); the *mechanical* half of accuracy—containment of the structured fields in the source text, what the model was shown, consistency with an external classification, agreement with a rule-based baseline, and the size of known false-positive classes—was computed in this revision for every mention or section without any new model call (Section 8.2); and the components that require independent human labels are specified, with the sampling frame drawn and weighted, but not executed (Section 8.3). Table 20 summarizes the status of the acceptance gates.

### 8.1. Coverage and integrity checks (implemented)

- **Universe reconciliation** (Table 2): the mapped universe and its two derived inputs are archived with checksums, and every mapped line is accounted for as matched or unmatched with the 7 unmatched tickers listed; the 65 issuers without filings are listed by name and reason. The source holdings file and the SEC ticker file are not archived and the lines dropped before mapping are not listed, so the check is partial (gate (a)).
- **Acquisition completeness:** for every issuer the SEC submissions index is enumerated in full (recent block and archive blocks) and every retained filing is logged; 2 filings logged as downloaded were not found on disk at parse time and are counted as failures. Filing counts by form and year are published (Table 26).
- **Parsing and scope coverage:** 16 of 527,548 primary documents could not be parsed (Table 3); filings without recognizable Item headings are kept as a single section rather than dropped, which for 10-Ks places them outside the submitted scope (35 filings invisible to the project layer); the 10-K funnel from parsed rows to ledgered rows reconciles exactly (Section 4.3).
- **Extraction ledger:** every ledgered section is recorded with its mention count and timestamp; the true number of calls (703,725) is recovered by recomputing the gate; the outage rows removed from the ledger reconcile with the run log and no row remains inside the outage gap (Section 5.5). The ledger does not record a per-call status, so an undetected API failure still appears as "no project": the run logs show 14 such failures outside the outage in October and 3 in June, and the first batch after the restart (5,198 sections, 23 productive against about 536 expected) is flagged as suspect pending a re-submission that was not authorised for this revision.
- **Two runs:** the June and October blocks can be separated only by the backfill timestamp; the ledger stores no model identifier or prompt hash, and no June code revision exists (Section 4.5). The inter-revision diff of the extractor and parser files is printed in Appendix E.
- **Referential integrity:** foreign keys from chunks to sections to filings to companies are enforced by the database; the merge of ten shards is append-only on globally unique keys and was checked for duplicate accession numbers and section identifiers (none). Mention identifiers, however, collide for 412 mentions in 304 sections (Section 5.4).
- **Reproducibility of exhibits:** every table, figure and inline number of this paper is regenerated from the exported data and audit outputs by two scripts in `code/`; 22 automated checks (`tests/`) verify that the exports reconcile with each other, that regeneration is byte-identical, and that the generator contains no data literal. The exports themselves were regenerated read-only on the master node in this revision and found byte-identical to the committed ones except for the ordering of tied rows in two files.

### 8.2. Mechanical accuracy audits (implemented in this revision)

**Field-level grounding.** The prompt requires that numbers and locations appear in the text, and the structured fields are literal tokens, so their containment in the stored section text can be tested for every mention (after case-folding, replacement of punctuation by spaces and a short alias list for country names, as stated in the note of Table 17; partners are the field most sensitive to this normalisation). For all 308,065 mentions joined to their section text (`results/grounding/`; 1,000 bootstrap replicates clustered by section), every listed year is found in 99.8% of mentions with years (99.7–99.8), every listed partner in 99.4%, the amount as a number (in any of the usual renderings) in 93.7% and with a currency token within forty characters in 80.5%, every listed location in 78.3% (10-K 85.7%, 10-Q 82.1%, 8-K 67.8%), and every technology tag in 63.6%; at the level of individual elements the rates are 99.7% for years, 99.6% for partners, 84.2% for locations and 70.6% for technologies. All structured fields of a mention are found in 88.2% of the 276,168 mentions that carry at least one (10-K 92.2%, 10-Q 92.4%, 8-K 76.6%; June block 84.5%, October 88.7%), with no difference between filings before and after the model's training cut-off beyond one percentage point. Three readings of these rates are needed. First, they are rates of *transcription*, not of correctness: a year, partner or amount can be copied faithfully from a passage that describes a counterparty's initiative, a hypothetical, or a financing instrument, so grounding is necessary for a correct mention and not sufficient. Second, the lower rates for locations and technologies reflect normalization by the model ("Texas" for "Austin, TX"; "cloud computing" for "AWS") as much as invention; the element-level rate for locations is 84.2%. Third, 604 mentions carry a year absent from the section, and the technology and year lists equal the regex output on the model's own description for up to 46% and 39% of mentions, so for those fields "grounded" may mean "copied from the paraphrase" (Section 5.4). The breakdown by field and form is in Table 17 and Figure 8; the full breakdown by item is in `results/grounding/grounding_by_field_form_item.csv`.

**Table 17.** Mechanical grounding of the structured fields of all stored mentions in the stored section text

| *Panel A: share of mentions whose field is found literally in the stored section text (%, 95% CI)* | | | | | |
|---|---|---|---|---|---|
| Field | Mentions with the field | All forms | 10-K | 10-Q | 8-K |
| Cited years | 252,469 | 99.8 [99.7, 99.8] | 99.6 | 99.8 | 99.8 |
| Partners | 81,320 | 99.4 [99.3, 99.5] | 99.3 | 99.1 | 99.6 |
| Amount (number) | 120,240 | 93.7 [93.6, 93.9] | 97.4 | 97.5 | 86.9 |
| Amount (number + currency token within 40 chars) | 120,240 | 80.5 [80.2, 80.7] | 86.8 | 82.3 | 74.9 |
| Locations | 118,783 | 78.3 [78.0, 78.6] | 85.7 | 82.1 | 67.8 |
| Technology tags | 36,740 | 63.6 [63.0, 64.2] | 63.8 | 64.8 | 58.2 |
| *Panel B: element level (every listed year, partner, location or tag counted separately)* | | | | | |
| Partners | 122,590 elements | 99.6 | | | |
| Locations | 181,898 elements | 84.2 | | | |
| Technology tags | 47,361 elements | 70.6 | | | |
| Cited years | 307,399 elements | 99.7 | | | |
| *Panel C: mentions with every structured field grounded (%), by form, run and confidence bin* | | | | | |
| All mentions with at least one structured field | 276,168 | 88.2 | 92.2 | 92.4 | 76.6 |
| By run: June (backfilled) / October | 36,806 / 239,362 | 84.5 / 88.7 | | | |
| confidence bin 0.45-0.6 | 696 | 90.7 | | | |
| confidence bin 0.6-0.7 | 1,726 | 93.5 | | | |
| confidence bin 0.7-0.8 | 13,313 | 94.2 | | | |
| confidence bin 0.8-0.9 | 58,586 | 92.7 | | | |
| confidence bin 0.9-1.0 | 201,847 | 86.4 | | | |
| confidence bin 10-K | 69,065 | 92.2 | | | |
| confidence bin 10-Q | 134,391 | 92.4 | | | |
| confidence bin 8-K | 72,712 | 76.6 | | | |

Population: all 308,065 stored mentions, joined to the text of their section as stored (legacy June text for the S&P 500 10-Ks). A field is "grounded" when every element of the list (years, partners, locations, technologies) or the amount (tested in units, thousands, millions and billions with one to three significant digits) occurs literally in that text; a currency token is $, USD, dollar, million or billion within 40 characters of the number. Before the search, element and text are normalised identically: case-folded, punctuation replaced by spaces, whitespace collapsed, and a short alias list applied to the element (U.S., USA → united states; U.K. → united kingdom; UAE; EU); the element must then occur as whole words. The normalisation matters for partners: on a reservoir sample of 20,000 mentions (5,287 with a partner), the rate is 99.3% under this rule and 96.1% under a plain case-folded substring search (`results/grounding/partners_normalisation_check.json`). Confidence intervals are percentile bootstraps with 1,000 replicates clustered by section. Grounding is a necessary condition for correctness, not a measure of it: a grounded year can belong to another initiative, and an ungrounded location can be a correct normalisation ("Texas" for "Austin"); precision and recall require the human labels of gate (d). Panel C counts mentions with at least one structured field. Upper bounds on the regex back-fill from the model's own paraphrase (`llm.py`): the stored years equal the regex output on the description for 98,438 of 252,469 mentions with years (39%) and the technology list for 16,949 of 36,740 (46%); 604 mentions cite a year absent from the section text.

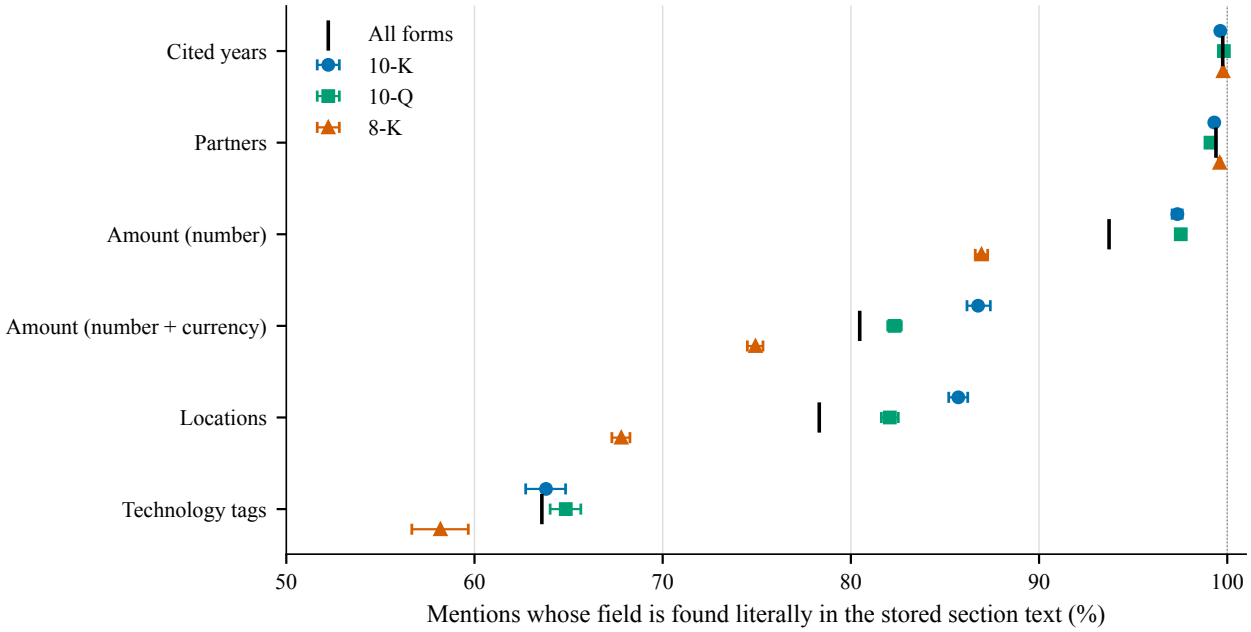



**Figure 8.** Mechanical grounding of the structured fields of all 308,065 stored mentions, by field and form. Markers: share of mentions whose field (every listed element, or the amount under its usual renderings) occurs literally in the stored section text; horizontal bars: 95% percentile-bootstrap intervals with 1,000 replicates clustered by section (narrower than the marker for most fields); black ticks: all forms pooled. The axis is truncated at 50% (no value lies below it). Grounding is a necessary condition for correctness, not a measure of precision (Table 17).

**What the model was shown.** Section 5.3 reports the shown fraction of every ledgered section. The recall of the pipeline on long sections is bounded from above by what the windows contain: on the over-budget sections the rule-based extractor of the code base recovers 26.2% of its full-text mentions from the focused text (20.8% in the 10-K MD&A). Gate (e) is therefore reported as met at passage level, not at section level.

**External consistency with 8-K item codes.** Form 8-K bodies name the items they report under, and 99.7% of the 400,601 8-K sections carry at least one item code in their text, which gives an external classification at no cost (`results/benchmark_8k/`). Among 8-Ks filed under Item 2.01 (completion of an acquisition or disposition; 5,456 sections) the model returned at least one `ma_integration` mention for 0.911 (0.903–0.918) of sections, against 0.199 under Item 8.01 (other events) and 0.031 under Item 5.02 (officer changes) and a base rate of 0.095; conversely 0.467 of 8-K `ma_integration` mentions sit in a filing under Item 2.01 or 1.01. Under Item 2.05 (exit or disposal costs; 1,700 sections) the model returned a `cost_reduction` mention for 0.709 (0.687–0.730) of sections, a lift of about 92 over the base rate, and 0.392 of `cost_reduction` mentions sit under that item; Item 1.01 (material agreements) carries a `partnership` mention in 0.040 of cases against a base rate of 0.011. These are consistency rates with an administrative

classification of the *filing*, not precision or recall of the *mention*: an Item 2.01 filing without an M&A mention may describe a disposition, and an Item 8.01 filing with one may be a genuine acquisition announced under a general item. They show that the dominant category is not an artefact of the extractor's vocabulary; they do not measure its accuracy (Table 18).

**Table 18.** External consistency of 8-K mentions with the 8-K item codes declared in the filing body

| Category | 8-K item code | Sections with the item | P(category \| item) %, 95% CI | P(item \| category) %, 95% CI | Base rate of the category (%) | Lift |
|---|---|---|---|---|---|---|
| M&A / integration | Item 2.01 | 5,456 | 91.1 [90.3, 91.8] | 13.1 [12.7, 13.4] | 9.5 | 9.6 |
| M&A / integration | Item 2.01 or 1.01 | 49,868 | 35.6 [35.2, 36.0] | 46.7 [46.2, 47.2] | 9.5 | 3.8 |
| M&A / integration | Item 1.01 | 46,888 | 31.8 [31.4, 32.2] | 39.2 [38.7, 39.7] | 9.5 | 3.3 |
| Cost reduction / restructuring | Item 2.05 | 1,700 | 70.9 [68.7, 73.0] | 39.2 [37.5, 41.0] | 0.8 | 92.1 |
| Strategic partnership | Item 1.01 | 46,888 | 4.0 [3.8, 4.2] | 42.1 [40.7, 43.6] | 1.1 | 3.6 |
| capex type | Item 1.01 | 46,888 | 13.2 [12.9, 13.6] | 46.8 [45.9, 47.6] | 3.3 | 4.0 |
| M&A / integration | Item 8.01 (control) | 92,840 | 19.9 [19.7, 20.2] | 48.7 [48.2, 49.2] | 9.5 | 2.1 |
| M&A / integration | Item 5.02 (control) | 71,781 | 3.1 [3.0, 3.2] | 5.9 [5.7, 6.1] | 9.5 | 0.3 |
| Cost reduction / restructuring | Item 8.01 (control) | 92,840 | 1.5 [1.4, 1.6] | 45.1 [43.3, 46.8] | 0.8 | 1.9 |
| M&A / integration | Item 2.03 (control) | 22,460 | 21.0 [20.5, 21.6] | 12.4 [12.1, 12.8] | 9.5 | 2.2 |
| Strategic partnership | Item 8.01 (control) | 92,840 | 1.7 [1.6, 1.8] | 34.9 [33.6, 36.4] | 1.1 | 1.5 |

Item codes are read from the 8-K body with the pattern `Item\s+([1-9]\.[0-9]2)` (99.7% of the 400,601 8-K sections carry at least one code). "P(category | item)" is the share of sections declaring the item in which the model returned at least one mention of the category; "P(item | category)" is the share of sections with such a mention that declare the item; lift is the first probability divided by the category's base rate over all 8-K sections. Item 2.01 (completion of an acquisition or disposition) and Item 2.05 (exit or disposal costs) are the natural counterparts of `ma_integration` and `cost_reduction`; Items 5.02 and 8.01 are controls. Confidence intervals are Wilson intervals. These are consistency proxies: an item code says that an event of the kind was reported, not that the model's mention is correct, so neither probability is a precision or a recall.

**Agreement with a rule-based baseline.** The code base contains a rule-based extractor (`rules.py`) that needs no model call. Run on a seeded, stratified sample of 30,000 productive and 30,000 zero-mention ledger sections with inverse-probability weights (`results/rules/`), the rules flag at least one candidate in 85.7% of the sections where the model returned a mention (10-K 97.4%, 10-Q 94.4%, 8-K 69.2%; June 79.8%, October 86.7%), while the model returns a mention in 32.5% of the sections the rules flag; the rules flag 39.5% of ledgered sections against 15.0% for the model, section-level agreement is 71.2% (model-only 2.1%, rules-only 26.6%), and on co-detected sections the model's category is among the rule categories in 84.6% of cases (`ma_integration` 96.8%, `drug_pipeline` 92.5%, `cost_reduction` 76.9%, `capex_program` 55.2%, `plant_construction` 40.7%, `geographic_expansion` 36.5%). Neither extractor is a ground truth, so this is a measure of the marginal information in the model step—it abstains on most of what the lexical rules flag and rarely fires where they do not—rather than of accuracy. A

full-corpus run of the rules is deferred; the sample gives roughly half a percentage point of precision on the rates (Table 19).

**Table 19.** Rule-based extractor versus LLM: section-level agreement on a stratified sample, and the recall bound of the focus windows

*Panel A: section-level agreement between the rule-based extractor and the LLM (weighted to the ledger)*

| Form | Sections sampled | P(rules $\geq 1$ \| LLM $\geq 1$) % | P(LLM $\geq 1$ \| rules $\geq 1$) % | Rules positive % | LLM positive % | Agreement % | Same type on co-detected % |
|---|---|---|---|---|---|---|---|
| 10-K | 8,890 | 97.4 | 28.2 | 73.9 | 21.4 | 46.4 | 86.2 |
| 10-Q | 29,399 | 94.4 | 28.7 | 43.5 | 13.3 | 68.2 | 87.8 |
| 8-K | 21,711 | 69.2 | 48.6 | 21.6 | 15.2 | 84.2 | 77.9 |
| All forms | 60,000 | 85.7 | 32.5 | 39.5 | 15.0 | 71.2 | 84.6 |

*Panel B: rule mentions recovered inside the focus windows, sections over the 9,000-character budget*

| Form | Sections over budget (sample) | Rule mentions, full text | Rule mentions, focused text | Recovered % |
|---|---|---|---|---|
| 10-K | 6,482 | 97,830 | 24,462 | 25.0 |
| 10-Q | 15,449 | 219,560 | 54,612 | 24.9 |
| 8-K | 2,997 | 6,847 | 5,767 | 84.2 |
| All forms | 24,928 | 324,237 | 84,841 | 26.2 |

Seeded stratified sample of 60,000 ledgered sections (30,000 with at least one LLM mention and the same number without), weighted back to the 1,123,098 ledger rows by inverse inclusion probability; the snapshot's `RuleExtractor` was run on the stored text. The two extractors have different constructs (the rules flag any gate-vocabulary sentence), so agreement is not accuracy and neither extractor is the reference. Panel B runs the rules on the full text and on the focused text of the same over-budget sections: the share recovered is a deterministic upper bound on what any extractor can see through the focus windows, not a recall estimate for the LLM.

**Known false-positive classes.** The financing-instrument pattern of Section 5.1 matches 7.3% of mentions, concentrated in Part II Item 2 of the 10-Q and in 8-K bodies, and Section 7 restates the category counts without them. A related parser check finds that the 10-Q Part II Item 2 section (median 602 words) is longer than 3,000 words in 1,001 of 77,869 filings (1.3%), 296 of which have an MD&A shorter than 500 words and are candidates for a Part I/Part II "Item 2" swap by the segmenter; 425 mentions come from those sections. Both classes are detected by pattern, not by annotation, so their sizes are indicative, and the annotation protocol labels them explicitly.

**Resolver sensitivity.** The dependence of the project count on the key and on the confidence threshold is reported in Section 6.2; the over-merge indicators (distinct names and partners per project, amounts taken from a different counterparty) are in Section 6.1.

### 8.3. Accuracy evaluation against human labels (frame drawn, labels not produced)

**Sampling frame.** The unit is the ledgered section, that is, exactly the text block the extractor received a verdict for. The frame was drawn with a fixed seed (20261105) from the two populations of the ledger (`results/annotation_frame/`; protocol in `docs/annotation_protocol.md`). From the 168,230 productive sections, 192 strata are defined by item group, run (June block or October), the shown-fraction bin of Section 5.3, the confidence bin of the section's best mention and the category group; allocation is proportional with a floor of 10 per stratum and a 2-fold oversampling of the confidence bin at or below 0.7, which gives $n = 1{,}677$ (the floor dominates; a nested core subsample of 510 drawn proportionally within the frame is flagged for a first pass). From the 954,868 zero-mention sections, 385 are drawn with inclusion probability proportional to the gate verdict and to the logarithm of section length, so that gate-rejected sections are represented (285 gate-passing, 100 gate-rejected) and false negatives can be estimated with inverse-probability weights; the earlier version's equal-count draw without weights would have biased recall upward. The frame is split into a development half (1,058 sections), on which prompt and threshold changes may be tuned, and a locked holdout half (1,004), used once per release. The design effects are 3.4 (productive) and 1.2 (zero-mention); the 95% half-widths are $\pm 4.4$ percentage points for a precision near 0.5 and $\pm 2.4$ points for a false-negative rate near 5%, and the minimum detectable difference in the false-negative rate between gate-passing and gate-rejected strata is 7.8 points. The full-text frames, the text the model was shown, and the sealed model output are stored on the master node (77 MB); the worktree copies carry the text truncated to 2,000 characters.

**Labels and annotators.** Two annotators will independently code each section without seeing the model output (it is unsealed only for the mention-level comparison): at the section level, whether the section describes at least one project and how many (labels S1–S2); at the mention level after unsealing, for each initiative its presence, category (from the 21-code taxonomy), status, cited years, location, partners, amount and amount type, plus the six-way label of Section 5.1 (project, generic strategy, third-party, financing instrument, uncertain, none) and a boilerplate flag (M1–M4); and at the project level, whether a cluster is correct, over-merged or under-merged (L1). Disagreements will be adjudicated by the third team member; inter-annotator agreement will be reported as Krippendorff's $\alpha$ and prevalence-adjusted $\kappa$ for presence and category, and as exact-match rates for fields. The annotators are the authors; no label has been produced at the time of writing, and no label generated by a language model will be reported as a human label.

**Metrics.** On the holdout, with the design weights: candidate-gate recall (share of annotated projects whose section passed the gate); project-detection precision, recall and F1 at the mention level, by form, item and shown-fraction bin; category macro- and micro-F1; field accuracy for

amount, years and location; the abstention rate (sections where the model returned an empty list) conditional on annotated content; the share of mentions in each false-positive class; and confidence calibration (precision by confidence bin, with bootstrap intervals, the 0.6 bin reported with and without the defaulted values). The same metrics will be computed for the rule-based extractor, so that the marginal value of the model step is documented, and the estimated error structure will be the input to the measurement-error correction that Ludwig et al. (2026) describe.

**Linkage audit.** From the resolved projects, a sample stratified by number of mentions (1, 2–5, 6–10, more than 10) and by category will be reviewed to label each project cluster as correct, over-merged (contains distinct initiatives) or under-merged (a duplicate of another cluster of the same issuer). Link precision and recall and the error costs of false merges versus missed links will be reported, with particular attention to long-running programs, subsidiaries, near-duplicate names and the generic-token keys of Section 6.1.

**Stability of the instrument.** A protocol of 500 sections re-submitted three times, once with a paraphrased prompt and once to a substitute model (about 2,500 calls), and a re-submission of the 2,686 gate-passing sections of the suspect post-outage batch, are specified; both require new model calls, which were not authorised for this revision, and their results are absent from this paper.

**Temporal test and missingness.** For a sample of projects with at least three mentions, the status and amount reconstructed using only mentions filed on or before each date will be compared with the full-information values, to verify that no later information leaks into earlier states and to document how often status reverses. Zero-project issuer-years will be cross-tabulated by sector, size and filing count to distinguish "no disclosure" from "not processed"; the former is informative, the latter is a data gap, and the `coverage_status` field (complete for the defined scope, partial, failed, not observed) proposed for the issuer-year export will carry that distinction.

### 8.4. Acceptance gates

**Table 20.** Acceptance gates for a research-ready release and their status at revision `2324b58` after the Stage D audits

| Gate | Requirement | Status |
|---|---|---|
| (a) | Target issuer universe and time window frozen and auditable | Partially met: snapshot design; mapped universe and derived inputs archived with SHA-256 checksums; the source holdings file and SEC ticker file are not archived and the lines dropped before mapping are not listed |
| (b) | Every published project links to at least one verifiable primary-source passage | Met at section level; structured fields mechanically checked against the section text (88.2% fully grounded); description fidelity not checkable; character spans not stored |
| (c) | Duplicate, temporal and amount rules documented and tested | Documented; the pipeline snapshot contains no test; the exhibit layer has 22 automated checks that do not test the pipeline; no event or amount-type tables; representative fields depend on tie order |
| (d) | Extraction and matching accuracy evaluated on independent human labels | Not met: protocol revised, frame drawn and weighted ($n = 1{,}677 + 385$), annotation pending (author task) |
| (e) | Issuer/form/year coverage and failures quantified | Met at passage level: coverage, parse and scope failures, gate rejections, outage, silent failures and shown fractions quantified; one post-outage batch flagged as suspect |
| (f) | Every empirical result regenerable from a pinned build | Exhibits and exports regenerable from the released code and data (`run_all.sh`, lock files); the databases are regenerable by the authors on the master node; the extraction step is re-runnable and ledgered, not deterministic; release of the derived tables is an author decision not yet taken |
| (g) | All coauthors have reviewed scientific claims and authorship | Pending |

Gates are those of the research-design specification (Section 3). "Met at passage level" means that the quantity is established for the text the model was actually shown, not for the whole section. Items marked as author tasks or decisions are listed in Section 10.1.

Until gates (c), (d) and (g) are met, the database is a *research database under validation*, and any use of its counts or amounts in empirical work should treat extraction and linkage error as measurement error of unknown structure, with the audits above as the only available bounds.

# 9. Research Applications and Identification Limits

## 9.1. What the database can support

Four families of empirical designs are natural once the validation gates are passed; each is listed with the restriction it must respect now.

**Project counts and composition as firm-year variables.** The issuer-year count of newly observed projects, by category, is a text-derived measure of disclosed strategic activity comparable in spirit to the thematic scores of the factor layer but with a discrete, interpretable unit. Because the panel is a snapshot universe, firm fixed effects and within-firm comparisons are safer than cross-sectional levels, and the later entry of smaller issuers (Figure 1) should be modeled explicitly. Two further restrictions follow from this revision: the count must be computed under a stated key and reported with its range (Section 6.2), and the S&P 500 cohort must carry a run indicator, since its mentions come from two runs and two 10-K segmenters and its per-issuer counts move by tens of projects depending on which run is counted (Section 7.3). A firm-year aggregation of mentions also requires the fiscal-year fill exported with the audit outputs, because `fiscal_year` is empty for 68,827 mentions (Section 7).

**Technology adoption and diffusion.** The `technologies` field and the categories `ai_initiative`, `data_center`, `cloud_migration`, `automation` and `erp_implementation` allow the first-mention date of a technology to be located by issuer, sector and location, in the spirit of Kalyani et al. (2025) but from the issuer's own regulatory narrative. The small counts of these categories (a few hundred projects each), the fact that technology tags are free text rather than a controlled vocabulary, and the possibility that up to 46% of tag lists were back-filled from the model's description (Section 5.4) limit precision; the adoption rates of the Census survey module on advanced technologies (Zolas et al., 2020) are an external benchmark against which the counts of these categories can be sanity-checked before use, and a mapping of tags to a vocabulary is part of the proposed ontology work.

**Event studies.** Mentions in 8-K filings are dated to the day and are the closest the database comes to an announcement date (74,566 stored mentions), in the tradition of the announcement-level evidence of McConnell and Muscarella (1985). Because the primary document of an 8-K often refers to a press release filed as an exhibit that the pipeline does not read, the first *public* announcement may precede the first mention in the corpus; event dates should be verified against the exhibit or the newswire before use, and the 19.2% of 8-K mentions that match the financing pattern should be excluded or relabeled first.

**Project-level panels.** For projects with several mentions, the timeline provides a sequence of status and amount statements that can be used to study announcement-to-completion durations, budget revisions and abandonment, subject to the caveats on status (last statement, not terminal fate; tie order for 972 projects), on merged phases (Section 6.1) and on amounts whose source wording, currency and scale are not stored (Section 6.4).

### 9.2. Identification limits

- **Disclosure, not activity.** The database measures what issuers chose to say in a regulated document. Disclosure thresholds, materiality judgments, boilerplate habits and litigation risk vary by firm, sector and time; a difference in project counts is a difference in disclosure before it is a difference in investment.
- **Snapshot universe and sampled proxy.** Survivorship and backfilling are built in, and the proxy omits an unknown subset of the smallest index members (Section 3). Comparisons across years that mix cohorts of issuers confound activity with the composition of the observed panel.
- **Measurement error of unknown structure.** Until the human-coded evaluation is run, extraction precision and recall, category accuracy and linkage error are unknown. The audits of Section 8 show that the error is unlikely to be classical: long sections are read only in part and in document order, so recall falls with section length and with the position of the passage; financing instruments and boilerplate-heavy items are more exposed to false positives; and the project count is partly an artefact of a key whose dominant branch is a generic token. Ludwig et al. (2026) set out the correction that a validation subsample permits; until it exists, the counts are descriptive quantities.
- **Run and cohort.** The S&P 500 block was extracted in a different run, with a different 10-K parser and an unknown model snapshot. Any design that compares that cohort with the rest of the universe, or that uses 10-K text features, must condition on the run.
- **Amounts.** Disclosed amounts are heterogeneous in type and scope, are never realized expenditure, and carry no source wording, currency or scale; the project-level maximum is a reading convenience, not an estimate, and the tail contains values that typing would most likely reject (Section 6.4).
- **Descriptive versus causal.** Associations between project disclosures and outcomes (returns, investment, employment) identify the joint determination of disclosure and outcomes, not the effect of the project. Causal designs require a source of variation in the *project* that is orthogonal to the decision to *describe* it, and the database does not supply one.

# 10. Limitations, Reproducibility and Conclusion

## 10.1. Limitations and what was not done

Beyond the identification limits above, the current build has concrete technical limitations that bound its use, and this revision has left several items unexecuted. Each is listed with its reason and, where one can be stated, the direction of the bias it implies.

- **Universe.** The issuer universe is a fund-holdings proxy at one date that covers 85.3% of the index count under representative sampling, not licensed Russell membership; the source holdings and ticker files were not archived, so the first line of Table 2 cannot be reconstructed; and 65 issuers are outside the form scope. The omitted members are expected to be the smallest, so cross-sectional statements about small issuers are the least reliable. A licensed constituent history would settle both points and was not available.
- **Parsing.** The parser stores section-level provenance but not character offsets into the raw HTML; tables are flattened in the corpus layer and were removed altogether from the S&P 500 10-K items prepared in June 2026; 35 10-K filings are invisible to the project layer. Re-parsing the S&P 500 10-Ks with the corpus-layer segmenter would make the two cohorts comparable and was not done because it would require re-extraction of their sections.
- **Focus windows.** The model saw a median 9.3% of a submitted 10-K MD&A and 62.7% of mentions come from sections shown for less than a quarter; recall on long sections is bounded above by the windows (26.2% of rule-based mentions survive them) and is lowest for passages late in a section, such as the liquidity and capital-resources discussion. Storing the focus spans and widening or re-ordering the windows are pipeline changes that imply new model calls.
- **Resolver.** The resolver is a deterministic key that both merges distinct phases and splits renamed initiatives; 54.8% of projects are keyed on a generic name token, the count ranges from 96,749 to 198,488 across reasonable keys, and the representative fields of 33,820 projects depend on tie order. The deterministic tie-breaker is specified and not yet adopted in the stored table (a pipeline change).
- **Extraction contract.** `amount_raw` is a formatted float, years and technologies may be back-filled from the model's paraphrase, missing confidences default to 0.6, out-of-vocabulary types default to `technology_deployment`, and the raw responses were not stored, so none of these can be undone without re-extraction. Amounts carry no type, currency or scale; the typing pass requires annotation or a second extraction and was not authorised.
- **Ledger.** The ledger cannot distinguish an empty result from a failed call: 14 silent failures are known from the October log, 3 from the June log, and the first post-outage batch (5,198 sections) is most likely a block of empty answers. The re-submission of its 2,686 gate-passing sections was requested and not granted for this revision; the affected issuers' October counts are biased

downward until it is run.

- **Two runs.** No June code revision exists, the ledger records no model identifier or prompt hash, and the June snapshot of the model is unknown; the runs are separable only by the backfill timestamp. This cannot be repaired after the fact; the proposed `extraction_runs` table prevents a recurrence.
- **Stability.** Temperature zero without a seed is not determinism; the re-submission, prompt-paraphrase and model-substitution protocol of Section 8.3 requires model calls that were not authorised, so the stability of the output is unmeasured.
- **Human labels.** The gold standard of Section 8.3 is drawn, weighted and documented but not annotated; precision, recall, category accuracy, calibration and linkage error therefore remain unmeasured, and gate (d) is not met. This is author labour and the main item of the next round.
- **Rule-based baseline.** The agreement rates of Section 8.2 are computed on a stratified sample of 30,000 productive and 30,000 zero-mention sections, not on the full ledger; a full-corpus run is a matter of compute time and was deferred.
- **Scope.** The 10-Q sections and 8-K bodies are read whole (subject to the windows) while 10-K Items 3–6 and 8–9B are not read at all, and 8-K exhibits (where press releases live) are not read; the first public announcement of a project may therefore precede its first mention.
- **Release and licence.** Whether the derived tables (`project_mentions`, `projects`, `project_timeline`, `extracted_sections`) are released as files with the paper, and under which licence, are author decisions not yet taken; until then gate (f) holds for the exports and exhibit code only.

### 10.2. Reproducibility, data availability and terms

**Data.** The primary documents are Forms 10-K, 10-Q and 8-K retrieved from the SEC's EDGAR system (public domain) under the fair-access policy, with an identifying user agent, between May 2026 (S&P 500 issuers) and October 27, 2026 (remaining issuers); they are not redistributed because of their size (254.5 GB) and can be re-downloaded with `scripts/download_filings.py`. The issuer universe merges the Wikipedia S&P 500 table scraped on May 25, 2026 (text under the Creative Commons Attribution-ShareAlike 4.0 licence) (Wikipedia contributors, 2026) and the iShares Russell 3000 ETF holdings file published by BlackRock, holdings as of October 23, 2026, downloaded on October 27, 2026 under the iShares website terms of use (BlackRock, Inc., 2026); the downloaded holdings file was not archived, while the mapped universe is archived with SHA-256 checksums. The index constituent count is read from the FTSE Russell factsheet of September 30, 2026 (FTSE Russell, 2026a). The extraction outputs were generated with the OpenAI API (alias `gpt-4o-mini` in June 2026; `gpt-4o-mini-2024-07-18` in October 2026 according to the authors' record) under OpenAI's terms of use; raw API responses were not stored, only the normalized

mentions. The derived tables reside in DuckDB on the master node of the authors' cluster and are available from the authors on request; their release as files with the paper is an author decision not yet taken.

**Code.** The acquisition, parsing and extraction pipeline is revision `2324b58` of the private repository `uqo-research-db.git`; the nine files that define the extraction instrument and the resolver are archived with the paper (`code/audit/spid_snapshot/`, with SHA-256 checksums). The exhibit code (`code/export_stats.py`, `code/make_tables.py`), the exports (`data/`), the audit scripts of this revision (`code/audit/`) and their outputs (`results/`) are released with the paper; `run_all.sh` regenerates every table, figure and inline number from the exports, and `tests/` (22 checks) verifies that the exports reconcile. The master-node environment is pinned (`requirements.lock`, 88 packages; Python 3.12.15; DuckDB 2.0.0); no licence file has been committed pending the authors' decision (MIT for code and CC BY 4.0 for derived tables are proposed).

**Reproducibility.** The exhibits are reproducible from the released materials. The databases are regenerable by the authors from the repository, the frozen universe artefact, the download logs and the extraction ledger; the paid extraction step is resumable and ledgered but not deterministic (Section 5.4), and the extraction cost about 775 USD at October 2026 prices (recorded against the October run; whether the total also covers the June block is to be confirmed by the authors). The repository contains no credentials and no private infrastructure details. The database does not contain personal data; the only names it stores are those of corporations, places and technologies as they appear in public filings.

### 10.3. Conclusion

We asked whether distinct strategic projects can be identified, classified, dated, linked and quantified from the SEC filings of a Russell 3000 proxy while preserving source evidence. We have documented a re-runnable, ledgered pipeline that turns 527,548 filings of 2,564 issuers into 308,065 section-linked candidate mentions and 121,099 candidate projects under one stated key, classified into a 21-category taxonomy and organized into timelines, and we have audited that pipeline mechanically: the structured fields are found verbatim in their source section for 88.2% of mentions, the model read a median 9.3% of a 10-K MD&A, the dominant category co-occurs with the corresponding 8-K item far above its base rate, 7.3% of mentions are financing instruments rather than projects, and the project count moves between 96,749 and 198,488 with the key. We have been equally explicit about what has *not* been done: the accuracy of extraction, classification and linkage has not been measured against independent human labels; the stability of the instrument has not been tested; amounts are not typed; events are not separated; one post-outage batch is suspect; the

universe is a sampled snapshot. The contribution of this paper is therefore the instrument and its audit rather than a validated measurement. The next step of the research program is the annotation of the frame delivered here and the linkage audit; only their results will determine whether the counts reported here can be used as variables, and with what measurement-error model. Until then, SPID is a research database under validation, and we ask that it be cited and used as such.

## References


**Afzal, Muhammad Talha, Jinqiu (Claire) Yan, Huaxia Li, and Miklos Vasarhelyi**, "Beyond the bottom line: using generative AI to extract narrative financial disclosures," *International Journal of Accounting Information Systems*, 2026, *57*, 100777.

**Bellstam, Gustaf, Sanjai Bhagat, and J. Anthony Cookson**, "A Text-Based Analysis of Corporate Innovation," *Management Science*, 2021, *67* (7), 4004–4031.

**Bettencourt, Nick, Xiaowei Ding, and Kay Giesecke**, "The Stanford EDGAR Filings Dataset: Reconstructing U.S. Corporate and Financial Disclosures into Layout-Faithful and Token-Efficient Pretraining Data," arXiv:2606.18192; also SSRN 7161699 (DOI 10.2139/ssrn.7161699) 2026.

**BlackRock, Inc.**, "iShares Russell 3000 ETF (IWV): Fund holdings file (`latest-holdings.csv`)," https://www.ishares.com/us/products/239714/ishares-russell-3000-etf 2026. Holdings as of October 23, 2026, downloaded October 27, 2026; subject to the iShares website terms of use; the downloaded file is not archived with the paper.

**Christen, Peter**, *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Berlin: Springer, 2012.

**Cohen, Lauren, Christopher Malloy, and Quoc Nguyen**, "Lazy Prices," *Journal of Finance*, 2020, *75* (3), 1371–1415.

**Crane, Leland D., Emily Green, Molly Harnish, Will McClennan, Paul E. Soto, Betsy Vrankovich, and Jacob Williams**, "Tracking Real Time Layoffs with SEC Filings: A Preliminary Investigation," Finance and Economics Discussion Series 2024-020, Board of Governors of the Federal Reserve System 2024.

**de Kok, Ties**, "ChatGPT for Textual Analysis? How to Use Generative LLMs in Accounting Research," *Management Science*, 2025, *71* (9), 7888–7906.

**Dolphin, Rian, Joe Dursun, Jarrett Blankenship, Katie Adams, and Quinton Pike**, "Grounded Event Extraction from SEC 8-K Filings with a Fine-Grained Taxonomy," arXiv:2607.08346; also SSRN 7087858 (DOI 10.2139/ssrn.7087858) 2026.

**Dyer, Travis, Mark Lang, and Lorien Stice-Lawrence**, "The Evolution of 10-K Textual Disclosure: Evidence from Latent Dirichlet Allocation," *Journal of Accounting and Economics*, 2017, *64* (2–3), 221–245.

**Fellegi, Ivan P. and Alan B. Sunter**, "A Theory for Record Linkage," *Journal of the American Statistical Association*, 1969, *64* (328), 1183–1210.

**FTSE Russell**, "Russell 3000 Index Factsheet," https://research.ftserussell.com/Analytics/FactSheets/Home/DownloadSingleIssue?issueName=US3000USD 2026. Index characteristics as of September 30, 2026 (2,996 holdings); downloaded November 5, 2026.

_ , "Russell US Equity Indexes: Construction and Methodology, v7.2," https://www.lseg.com/content/dam/ftse-russell/en_us/documents/ground-rules/russell-us-indexes-construction-and-methodology.pdf August 2026. Document dated August 2026; downloaded November 5, 2026.

**Gentzkow, Matthew, Bryan Kelly, and Matt Taddy**, "Text as Data," *Journal of Economic Literature*, 2019, *57* (3), 535–574.

**Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli**, "ChatGPT outperforms crowd workers for text-annotation tasks," *Proceedings of the National Academy of Sciences*, 2023, *120* (30), e2305016120.

**Greenstone, Michael, Richard Hornbeck, and Enrico Moretti**, "Identifying Agglomeration Spillovers: Evidence from Winners and Losers of Large Plant Openings," *Journal of Political Economy*, 2010, *118* (3), 536–598.

**Grimmer, Justin and Brandon M. Stewart**, "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts," *Political Analysis*, 2013, *21* (3), 267–297.

**Hassan, Tarek A., Stephan Hollander, Laurence van Lent, and Ahmed Tahoun**, "Firm-Level Political Risk: Measurement and Effects," *Quarterly Journal of Economics*, 2019, *134* (4), 2135–2202.

**Hoberg, Gerard and Gordon Phillips**, "Text-Based Network Industries and Endogenous Product Differentiation," *Journal of Political Economy*, 2016, *124* (5), 1423–1465.

_ **and Vojislav Maksimovic**, "Redefining Financial Constraints: A Text-Based Analysis," *Review of Financial Studies*, 2015, *28* (5), 1312–1352.

**Jha, Manish, Jialin Qian, Michael Weber, and Baozhong Yang**, "ChatGPT and Corporate Policies," NBER Working Paper 32161, National Bureau of Economic Research 2024.

**Kalyani, Aakash, Nicholas Bloom, Marcela Carvalho, Tarek Hassan, Josh Lerner, and Ahmed Tahoun**, "The Diffusion of New Technologies," *Quarterly Journal of Economics*, 2025, *140* (2), 1299–1365. Published version of NBER Working Paper 28999 (2021, DOI 10.3386/w28999).

**Kammoun, Manel, Djerrycharli Tandjambianda, and Simon-Pierre Boucher**, "Does the Language of Artificial Intelligence Move Markets? A Semantic Analysis of Russell 3000 SEC Filings and the Cross-Section of Stock Returns, 2010–2026," UQO Working Paper (FIRC) 0002, Université du Québec en Outaouais, Département des sciences administratives 2026. Version 2.0, October 28, 2026.

**Kelly, Bryan, Dimitris Papanikolaou, Amit Seru, and Matt Taddy**, "Measuring Technological Innovation over the Long Run," *American Economic Review: Insights*, 2021, *3* (3), 303–320.

**Kim, Alex G., Maximilian Muhn, and Valeri V. Nikolaev**, "From Transcripts to Insights: Uncovering Corporate Risks Using Generative AI," Working paper 4593660, SSRN 2023.

**Li, Kai, Feng Mai, Rui Shen, and Xinyan Yan**, "Measuring Corporate Culture Using Machine Learning," *Review of Financial Studies*, 2021, *34* (7), 3265–3315.

**Li, Yuliang, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan**, "Deep entity matching with pre-trained language models," *Proceedings of the VLDB Endowment*, 2020, *14* (1), 50–60.

**Loughran, Tim and Bill McDonald**, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks," *Journal of Finance*, 2011, *66* (1), 35–65.

_ **and** _ , "Textual Analysis in Accounting and Finance: A Survey," *Journal of Accounting Research*, 2016, *54* (4), 1187–1230.

**Loukas, Lefteris, Manos Fergadiotis, Ion Androutsopoulos, and Prodromos Malakasiotis**, "EDGAR-CORPUS: Billions of Tokens Make The World Go Round," in "Proceedings of the Third Workshop on Economics and Natural Language Processing" 2021, pp. 13–18.

**Ludwig, Jens, Sendhil Mullainathan, and Ashesh Rambachan**, "Large Language Models: An Applied Econometric Framework," *Annual Review of Economics*, 2026, *18* (1), 283–316. Also NBER Working Paper 33344 (2025, DOI 10.3386/w33344).

**McConnell, J and Chris J. Muscarella**, "Corporate capital expenditure decisions and the market value of the firm," *Journal of Financial Economics*, 1985, *14* (3), 399–422.

**Peeters, Ralph, Aaron Steiner, and Christian Bizer**, "Entity Matching using Large Language Models," arXiv:2310.11244 2023.

**Raasveldt, Mark and Hannes Mühleisen**, "DuckDB: An Embeddable Analytical Database," in "Proceedings of the 2019 International Conference on Management of Data (SIGMOD)" 2019, pp. 1981–1984.

**Reimers, Nils and Iryna Gurevych**, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)" 2019, pp. 3982–3992.

**Sautner, Zacharias, Laurence van Lent, Grigory Vilkov, and Ruishen Zhang**, "Firm-Level Climate Change Exposure," *Journal of Finance*, 2023, *78* (3), 1449–1498.

**Tandjambianda, Djerrycharli, Simon-Pierre Boucher, and Manel Kammoun**, "SPID — SEC Project Intelligence Database: A Methodological Report on the S&P 500 Release," Unpublished internal methodological report, available from the authors, Université du Québec en Outaouais, Département des sciences administratives 2026. June 2026 study of revision `df7905d`; the report file was regenerated on October 28, 2026.

**U.S. Securities and Exchange Commission**, "Accessing EDGAR Data: Fair Access Policy," https://www.sec.gov/search-filings/edgar-search-assistance/accessing-edgar-data 2026. Accessed October 2026; URL re-verified November 5, 2026 (the former `/os/accessing-edgar-data` address redirects here).

**Wikipedia contributors**, "List of S&P 500 companies," https://en.wikipedia.org/wiki/List_of_S%26P_500_companies 2026. Table scraped May 25, 2026; text available under the Creative Commons Attribution-ShareAlike 4.0 licence.

**Yang, Daniel, Yao-Hung Hubert Tsai, and Makoto Yamada**, "On Verbalized Confidence Scores for LLMs," arXiv:2412.14737 2024.

**Ye, Fanghua, Mingming Yang, Jianhui Pang, Longyue Wang, Derek Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu**, "Benchmarking LLMs via Uncertainty Quantification," in "Advances in Neural Information Processing Systems 37 (NeurIPS 2024)" 2024, pp. 15356–15385.

**Zolas, Nikolas, Zachary Kroff, Erik Brynjolfsson, Kristina McElheran, David Beede, Cathy Buffington, Nathan Goldschlag, Lucia Foster, and Emin Dinlersoz**, "Advanced Technologies Adoption and Use by U.S. Firms: Evidence from the Annual Business Survey," NBER Working Paper 28290, National Bureau of Economic Research 2020.

# Appendices

## A. Taxonomy

**Table 21.** The 21-category SPID project taxonomy

| Code | Label |
|---|---|
| manufacturing_expansion | Manufacturing / Capacity Expansion |
| plant_construction | Plant / Site Construction |
| data_center | Data Center |
| cloud_migration | Cloud Migration |
| ai_initiative | AI / ML Initiative |
| digital_transformation | Digital Transformation |
| erp_implementation | ERP / Systems Implementation |
| rd_program | R&D Program |
| product_launch | New Product / Launch |
| drug_pipeline | Drug / Clinical Pipeline |
| ma_integration | M&A / Integration |
| geographic_expansion | Geographic Expansion |
| capex_program | Capital Expenditure Program |
| supply_chain | Supply Chain Initiative |
| automation | Automation / Robotics |
| sustainability | Sustainability / ESG |
| energy_transition | Energy Transition / Renewables |
| infrastructure | Infrastructure Investment |
| cost_reduction | Cost Reduction / Restructuring |
| partnership | Strategic Partnership |
| technology_deployment | Technology Deployment |

Codes and labels read from `spid/taxonomy.py` of the frozen snapshot (`code/audit/spid_snapshot/`, SHA-256 in `SHA256SUMS`); the labels are passed verbatim to the model in the order shown (Appendix B). A label outside the list is mapped by substring to the closest code or, failing that, to `technology_deployment`.

## B. Extraction contract

The system prompt below is reproduced from `spid/extract/llm.py` with the 21 taxonomy codes and labels substituted exactly as the code does (generated by `code/make_tables.py`; never

retyped). The user message is the string "SECTION: <section_name>" followed by the focused section text (Section 5.3), cut at 14,000 characters. Decoding parameters: temperature 0, JSON-object response format (valid JSON is guaranteed; the field list is not enforced by a schema), 2,000 completion tokens, three attempts before an empty list is returned, 120-second timeout, no seed. The model identifier is not recorded in the ledger or the logs (Section 4.5). Straight quotes are rendered as sent.

```
You are a financial analyst extracting CORPORATE PROJECTS from SEC filing text.
A "project" is any concrete initiative, investment, program, expansion, construction,
product/R&D effort, acquisition/integration, technology deployment, partnership, or
strategic program the company is pursuing.

Return STRICT JSON: {"projects": [ ... ]}. Each project object has:
  project_name  : short descriptive name (use the official name if one is stated)
  project_type  : EXACTLY ONE of the allowed types below
  description   : 1 sentence, grounded in the text (no speculation)
  objective     : the goal/why (or "")
  status        : one of planned | in_progress | completed | mentioned
  amount_usd    : numeric USD figure if stated (e.g. 1500000000), else null
  locations     : array of places (city/state/country), else []
  technologies  : array of technologies involved, else []
  partners      : array of named partners/JV counterparties, else []
  benefits      : array of expected benefits, else []
  risks         : array of stated risks, else []
  years         : array of 4-digit years mentioned, else []
  confidence    : 0.0-1.0 (how clearly this is a real, concrete project)

Allowed project_type values:
  - manufacturing_expansion: Manufacturing / Capacity Expansion
  - plant_construction: Plant / Site Construction
  - data_center: Data Center
  - cloud_migration: Cloud Migration
  - ai_initiative: AI / ML Initiative
  - digital_transformation: Digital Transformation
  - erp_implementation: ERP / Systems Implementation
  - rd_program: R&D Program
  - product_launch: New Product / Launch
  - drug_pipeline: Drug / Clinical Pipeline
  - ma_integration: M&A / Integration
  - geographic_expansion: Geographic Expansion
  - capex_program: Capital Expenditure Program
  - supply_chain: Supply Chain Initiative
  - automation: Automation / Robotics
```

```
  - sustainability: Sustainability / ESG
  - energy_transition: Energy Transition / Renewables
  - infrastructure: Infrastructure Investment
  - cost_reduction: Cost Reduction / Restructuring
  - partnership: Strategic Partnership
  - technology_deployment: Technology Deployment

Rules:
- Extract ONLY real initiatives, not generic risk-factor boilerplate or hypotheticals.
- If the text describes no concrete project, return {"projects": []}.
- Do not invent facts. Numbers/locations must appear in the text.
- Prefer fewer high-quality projects over many vague ones.
```

The candidate gate is the case-insensitive regular expression below, reproduced from `spid/extract/rules.py` (25 alternatives, 215 characters, compiled with `re.IGNORECASE`); line breaks are inserted for display only. Sections without a match, or with fewer than 30 words, are not submitted; sections with fewer than 30 words are not ledgered either, except the 1,070 June rows backfilled without the filter.

```
\b(project|initiative|program|invest|expan|construct|build|launch|develop|
deploy|implement|acqui|partner|facilit|capacity|capital|migrat|transform|
pipeline|plant|data ?cent|automat|sustainab|renewable|restructur)\w*
```

## C. Data dictionary and data-model audit

Table 22 lists the tables required by the research-design specification against the current implementation (generated from `data/datamodel_audit.csv`). The full field-level dictionary of the current tables is maintained in `docs/data_dictionary.md` of the working-paper folder and in the repository documentation; `amount_raw` is documented there as a formatted copy of `amount_usd`, not as a source string.

**Table 22.** Tables required by the research design, current implementation and status

| Required table | Current implementation (database, table, key fields) | Status |
|---|---|---|
| `universe_snapshots` | `data/metadata/universe.json` + `companies` (CIK, ticker(s), name, sector, flags `in_sp500`/`in_russell3000`, fund weight, SIC, state, exchange, FYE, shard); snapshot date and source URL recorded | Partial: no file hash, no membership dates, no match confidence |

*continued on the next page*

*Table 22 (continued)*

| Required table | Current implementation (database, table, key fields) | Status |
|---|---|---|
| `issuers` | `companies` (one row per CIK; former names from SEC header stored in metadata JSON only) | Partial: no name history, no parent/subsidiary links |
| `filings` | `filings` (accession PK, CIK FK, form, filing/report dates, SEC URL, local path, size, download and parse status) | Implemented; no acceptance timestamp, no checksum |
| `sections` | `filing_sections` (UUID PK, accession FK, name, item, order, text, counts) | Implemented; no character offsets into raw HTML |
| `project_mentions` | `project_mentions` (SHA-1 id, accession, CIK, section id, name, type, description, objective, amount, formatted amount string (`amount_raw`, not the filing's string), lists, years, status, confidence, backend, issuer context) | Implemented; no evidence span, no prompt hash, no five-way label |
| `projects` | `projects` (MD5 of CIK\|type\|location-or-name, representative fields, max amount, counts, first/last seen and cited years) | Implemented (deterministic key) |
| `project_mention_links` | implicit in the key (every mention maps to exactly one project) | Proposed: explicit table with score, resolver version, adjudication |
| `project_events` | `project_timeline` (one row per mention: date, form, status, amount, confidence, snippet) | Partial: mention rows, not typed events |
| `project_amounts` | `amount_usd`/`amount_raw` on mentions; max on projects | Proposed: type, currency, scale, basis, realization flag |
| `extraction_runs` | `extracted_sections` ledger (section, accession, backend, n_mentions, timestamp) + run logs | Partial: no prompt/schema/taxonomy hashes, no call status |
| `human_labels` | — | Proposed |

Status at revision `2324b58`; "partial" and "proposed" rows are schema changes to be added by migration. The `amount_raw` field is a formatted copy of the coerced float, not the filing's string; the ledger stores no model, prompt or call status.

# D. Evidence register

**Table 23.** Evidence register: status of the main claims of this paper at revision `2324b58` after the Stage D audits

| Claim | Status |
|---|---|
| Universe of 2,564 issuers built from the IWV holdings file (October 23, 2026) and the S&P 500 cache (May 25, 2026); covers 85.3% of the index constituent count | Implemented; derived artefacts verified and hashed; source holdings and ticker files not archived; index count read from the factsheet |
| 527,548 primary documents of Forms 10-K/10-Q/8-K since 2010 downloaded; 16 not parsed (2 missing on disk) | Implemented, verified (database counts; accession list published) |
| Acquisition respected the SEC rate limit (aggregate token bucket; no 403 recorded) | Implemented; documented from run notes; orchestrator logs not archived |
| 1,342,283 sections, 10,720,890 chunks and embeddings; factor panels for 2,499 issuers | Implemented, verified (database counts) |
| 1,123,098 sections ledgered; 703,725 calls; outage rows removed reconcile with the log | Implemented, verified (gate recomputed; log reconciled) |
| Ledger free of failed-call artefacts | Not verified: 14 silent failures outside the outage; first post-outage batch (5,198 sections) suspect; re-submission not authorised |
| Two runs under one prompt text | File-system evidence only (modification times, two-line diff between October revisions); no June revision, no model identifier or prompt hash in the ledger |
| 308,065 stored mentions (308,477 returned; 412 lost to identifier collisions), 121,099 projects under the current key, graph of 183,472 nodes | Implemented, verified; project count depends on the key (96,749–198,488); representative fields depend on tie order |
| Structured fields of each mention grounded in its source section | Measured for all mentions: 88.2% fully grounded (years 99.8%, partners 99.4%, amounts 93.7%, locations 78.3%, technologies 63.6%); transcription, not correctness |
| Description grounded in the passage | Documented (prompt rule); not mechanically checkable (no span stored) |
| What the model was shown | Measured for every ledgered section (median 9.3% of a 10-K MD&A; 62.7% of mentions from sections shown $< 25\%$); recall bound 26.2% |
| Category consistent with an external classification | Measured for 8-Ks as a consistency proxy (Item 2.01 $\rightarrow$ `ma_integration` 0.911; Item 2.05 $\rightarrow$ `cost_reduction` 0.709); not precision or recall |

*continued on the next page*

*Evidence register (continued)*

| Claim | Status |
|---|---|
| Agreement with a rule-based baseline | Measured on a weighted sample of 30,000 + 30,000 sections (rules flag 85.7% of model-productive sections); neither is a ground truth |
| Financing instruments and Item 2 swaps as false-positive classes | Measured by pattern (7.3% of mentions; 296 swap candidates); not annotated |
| Extraction precision/recall, category accuracy, field accuracy, calibration | Not measured; frame drawn and weighted ($n = 1{,}677 + 385$), protocol published, labels pending (author task) |
| Linkage precision/recall of the resolution key | Not measured; over-merge indicators measured (Section 6.1); audit protocol specified |
| Stability under re-submission, prompt paraphrase and model substitution | Not measured; protocol specified; requires model calls not authorised |
| Amounts typed and separated from realized expenditure | Not implemented (proposed); `amount_raw` is a formatted float; tail audited (161 mentions $\geq$ 50 B USD) |
| Dated project events and point-in-time revision flags | Not implemented (mention-level timeline only) |
| Release of the derived tables; licence | Author decisions pending |
| Author order, affiliations and CRediT contributions | Pending confirmation by the three coauthors |

Notes: "Measured" entries refer to the read-only audits of `code/audit/` whose outputs are in `results/` with a manifest (script hash, database sizes and modification times, row counts); none involved a model call. "Verified" means recomputed from the exported data or the databases; "documented" means stated by the code or the run notes without an independent check.

## E. The two extraction runs

**Git and file-system evidence.** The repository holds six commits, all dated October 27–28, 2026 (`results/runs/git_evidence.md`); `project-db/` enters at the first commit. The extractor modules carry modification times of June 17, 2026 and SHA-256 hashes recorded in `code/audit/spid_snapshot/SHA256SUMS`; the driver files were edited on October 27, 2026 before the first commit. The diff between the revision the earlier version called "the June revision" (`df7905d`, dated October 27) and the build revision `2324b58` on the extractor, parser and driver files is the following change to `store.py`:

```
diff --git a/project-db/spid/store.py b/project-db/spid/store.py
index 9b2d7d1..5265333 100644
--- a/project-db/spid/store.py
+++ b/project-db/spid/store.py
```

```
@@ -186,7 +186,9 @@ class Store:
         sql += " ORDER BY filing_date"
         if limit:
             sql += f" LIMIT {int(limit)}"
-        return self.con.execute(sql).fetch_arrow_table().to_pylist()
+        res = self.con.execute(sql)
+        table = res.to_arrow_table() if hasattr(res, "to_arrow_table") else res.fetch_arrow_table()  # DuckDB
      >= 2.0 / < 2.0
+        return table.to_pylist()

     # ---- write side ----------------------------------------------------
     def upsert_mentions(self, mentions: list[ProjectMention], ctx: dict) -> int:
```

The diff from the first commit (0304133) to 2324b58 on the same files adds the following change to the corpus-layer `filing_parser.py` (the patch repeats the `store.py` lines above):

```
diff --git a/project-db/spid/store.py b/project-db/spid/store.py
index 9b2d7d1..5265333 100644
--- a/project-db/spid/store.py
+++ b/project-db/spid/store.py
@@ -186,7 +186,9 @@ class Store:
         sql += " ORDER BY filing_date"
         if limit:
             sql += f" LIMIT {int(limit)}"
-        return self.con.execute(sql).fetch_arrow_table().to_pylist()
+        res = self.con.execute(sql)
+        table = res.to_arrow_table() if hasattr(res, "to_arrow_table") else res.fetch_arrow_table()  # DuckDB
      >= 2.0 / < 2.0
+        return table.to_pylist()

     # ---- write side ----------------------------------------------------
     def upsert_mentions(self, mentions: list[ProjectMention], ctx: dict) -> int:
diff --git a/src/parsing/filing_parser.py b/src/parsing/filing_parser.py
index 1d9f6cc..e1327ed 100644
--- a/src/parsing/filing_parser.py
+++ b/src/parsing/filing_parser.py
@@ -18,6 +18,7 @@ from __future__ import annotations

 import json
 import traceback
+import uuid
 from concurrent.futures import ProcessPoolExecutor, as_completed
 from datetime import datetime, timezone
 from pathlib import Path
@@ -78,6 +79,18 @@ class FilingParser:
             return self._error_result(accession_number, form_type, "empty_after_clean")

         sections = self._extractor.extract_sections(clean_text, form_type)
+        if not sections and clean_text:
+            # no Item markers found (e.g. exhibit-only 10-Q/10-K): keep the whole text as one
+            # real section so the chunks satisfy the filing_chunks -> filing_sections foreign key
+            sections = [{
+                "section_id": str(uuid.uuid4()),
+                "section_name": "full_document",
+                "section_order": 0,
```

```
+                "item_number": "N/A",
+                "text": clean_text.strip(),
+                "word_count": len(clean_text.split()),
+                "char_count": len(clean_text),
+            }]
         for section in sections:
             section["accession_number"] = accession_number

@@ -88,13 +101,6 @@ class FilingParser:
                 chunk["accession_number"] = accession_number
             chunks.extend(section_chunks)

-        if not chunks and clean_text:
-            fallback_chunks = self._chunker.chunk_text(clean_text)
-            for chunk in fallback_chunks:
-                chunk["section_id"] = "full_document"
-                chunk["accession_number"] = accession_number
-            chunks.extend(fallback_chunks)
-
         result = {
             "accession_number": accession_number,
             "form_type": form_type,
```

**Mentions and projects by cohort, run and form.** Table 24 reconciles the mentions of the S&P 500 cohort (`in_sp500` true) and of the Russell-only cohort by run and form; the June block is identified by the backfill timestamp. The S&P 500 cohort's 10-Q mentions come from both runs for the same issuers, which permits a within-issuer comparison. Pooled over the cohort's 10-Q ledger rows, the productive share is 12.4% in June and 12.7% in October; on the 81 issuers with 10-Qs in both runs, the per-issuer means are 11.9% and 12.5% (paired difference 0.6 percentage points, standard error 1.3). The cohort is the same in both files (`in_sp500` of the issuer), but Table 24 counts stored mentions whereas the within-cohort files sum the ledger's mention counts, which exceed the stored ones by the identifier collisions of Section 5.5. Mentions per row are 0.244 and 0.250, mean confidence 0.889 and 0.893, and the M&A share 49.9% and 45.7% (`results/runs/within_sp500_*.csv`). For 10-Qs, whose text went through the same parser in both runs, the two runs therefore behave alike; the cohort difference of Section 7.3 comes from the volume of sections and from the 10-K segmenter, not from the extractor.

**Table 24.** Mentions by cohort, run and form: reconciliation of the June (S&P 500) and October (Russell proxy) runs

| Cohort | Run | Form | Mentions | Projects touched | Filings | Issuers |
|---|---|---|---|---|---|---|
| Russell proxy only | October run | 10-K | 64,798 | 34,658 | 15,784 | 1,857 |
| Russell proxy only | October run | 10-Q | 117,224 | 45,274 | 40,745 | 1,954 |
| Russell proxy only | October run | 8-K | 55,676 | 34,420 | 45,937 | 1,964 |
| S&P 500 | June run (backfilled) | 10-K | 15,787 | 8,697 | 4,193 | 481 |
| S&P 500 | June run (backfilled) | 10-Q | 5,817 | 2,071 | 2,028 | 84 |
| S&P 500 | June run (backfilled) | 8-K | 18,326 | 10,023 | 15,234 | 495 |
| S&P 500 | October run | 10-K | 910 | 649 | 222 | 97 |
| S&P 500 | October run | 10-Q | 28,963 | 11,030 | 10,684 | 455 |
| S&P 500 | October run | 8-K | 564 | 432 | 477 | 203 |

The June run covered the 10-Ks (legacy segmenter), the 8-Ks and 3,710 10-Qs of the S&P 500; the October run added the remaining S&P 500 10-Qs and the whole Russell-only cohort. "Projects touched" counts distinct current-key projects with at least one mention in the row, so projects spanning rows are counted more than once. The backfilled block alone resolves to the June release's project count, which reconciles the two releases without any resolver change.

**Fiscal year.** `fiscal_year` is empty for 68,827 mentions because the corresponding 134,124 filings carry no fiscal year in the corpus layer; a derived fill from the report or filing date is exported for 31,976 accessions (`results/runs/fiscal_year_fill_by_accession.csv`) and is not stored in the database.

# F. Additional tables

**Table 25.** Universe artefacts: file digests computed on the master node and release status

| Artefact | SHA-256 | Bytes | Shipped with the paper |
|---|---|---|---|
| `universe.json` | `3967b29b378e50fbbac139bfc9051207 49ff09dcf8ec5769c3fbb12cb1c016e2` | 839,099 | yes (data/metadata/) |
| `russell3000_ constituents.json` | `758cbebf58540230ede17fd35890336b a6bf9e29fad58e3792ce7134997c6b41` | 751,897 | yes (data/metadata/) |
| `sp500_constituents. json` | `5084075260692130896c3c9a187223e2 870b8e73774b35d1e28763fe8ffddb7d` | 109,110 | yes (data/metadata/) |
| `cik_mapping.json` | `7ec59b276cef779f8f714d29d7b77938 ee3731f11ea5bd95526006fd03ed20fc` | 481,386 | no (hash only) |
| `shards.json` | `8351ff54bea877175ca4bba51fc043ba 279b2bfbcc7ddb80db1da534ff922771` | 57,411 | no (hash only) |
| FTSE Russell factsheet (2026-09-30) | `281e92643781cf19a3f4bc6a670e0719 aa5ef1347d1071300491274aed0436e4` | – | no (public URL) |
| Russell US indexes methodology (v7.2) | `26d2d051beb27a6a808af7c17a2e2871 cc5388e669bedbb42cbc4d0d3ce47c76` | – | no (public URL) |

Digests computed read-only on the master node (`results/universe/SHA256SUMS`); the three constituent files are shipped under `data/metadata/`. The iShares holdings CSV and the SEC `company_tickers.json` consumed by the build were searched for on the node and not found (none): gate (a) of Table 20 is met for the frozen JSON artefacts only, not for their upstream sources. `cik_mapping.json` and `shards.json` are hashed but not released (the latter names private cluster nodes); the URLs of the two public FTSE Russell documents are recorded in `results/russell_coverage.json`.

**Table 26.** Issuers, filings, stored mentions and projects first seen, by filing year

| Year | Issuers filing | 10-K | 10-Q | 8-K | Mentions stored | Issuers with $\geq 1$ mention | Mentions per issuer with mention | Projects first seen |
|---|---|---|---|---|---|---|---|---|
| 2010 | 1,392 | 1,357 | 4,152 | 17,672 | 11,908 | 1,224 | 9.7 | 7,835 |
| 2011 | 1,424 | 1,409 | 4,260 | 18,533 | 13,551 | 1,287 | 10.5 | 6,679 |
| 2012 | 1,472 | 1,436 | 4,363 | 19,437 | 14,302 | 1,333 | 10.7 | 6,444 |
| 2013 | 1,539 | 1,483 | 4,530 | 19,773 | 14,611 | 1,408 | 10.4 | 6,286 |
| 2014 | 1,619 | 1,549 | 4,780 | 20,927 | 16,274 | 1,490 | 10.9 | 6,732 |
| 2015 | 1,689 | 1,635 | 4,983 | 22,077 | 17,073 | 1,535 | 11.1 | 6,675 |
| 2016 | 1,744 | 1,692 | 5,146 | 22,447 | 16,429 | 1,579 | 10.4 | 6,240 |
| 2017 | 1,810 | 1,745 | 5,359 | 22,897 | 17,350 | 1,636 | 10.6 | 6,490 |
| 2018 | 1,893 | 1,827 | 5,582 | 23,709 | 18,064 | 1,737 | 10.4 | 6,815 |
| 2019 | 1,973 | 1,903 | 5,829 | 24,256 | 18,737 | 1,806 | 10.4 | 6,837 |
| 2020 | 2,089 | 1,985 | 6,095 | 27,424 | 17,970 | 1,919 | 9.4 | 7,212 |
| 2021 | 2,247 | 2,116 | 6,606 | 27,944 | 21,312 | 2,077 | 10.3 | 8,216 |
| 2022 | 2,291 | 2,258 | 6,830 | 26,533 | 22,720 | 2,105 | 10.8 | 8,158 |
| 2023 | 2,329 | 2,295 | 6,938 | 26,851 | 22,955 | 2,129 | 10.8 | 7,977 |
| 2024 | 2,387 | 2,341 | 7,092 | 27,438 | 22,503 | 2,204 | 10.2 | 7,819 |
| 2025 | 2,453 | 2,406 | 7,300 | 28,716 | 23,220 | 2,280 | 10.2 | 7,990 |
| 2026 | 2,498 | 2,338 | 5,322 | 23,972 | 19,086 | 2,260 | 8.4 | 6,694 |

Filing year of the document. 2010 is left-censored (a project already under way before the corpus start is "first seen" at its first in-window filing) and 2026 is partial (filings to 2026-10-27); the range quoted in the text, 6,240–8,216 projects first seen per year, covers 2011–2025 only. "Projects first seen" counts projects under the current resolution key whose earliest mention falls in the year (Table 27 holds the issuer-cohort and windowed-key series). Issuers filing rise mechanically under the snapshot design (Section 3).

**Table 27.** Mentions and projects first seen by filing year: all issuers versus a fixed issuer cohort and windowed keys

| | All issuers filing in the year | | | | | Fixed cohort: issuers filing in every year 2010–2025 | | | |
|---|---|---|---|---|---|---|---|---|---|
| Year | Issuers | Mentions per issuer | First seen (current key) | First seen ($k = 3$) | First seen ($k = 5$) | Issuers | Mentions per issuer | First seen (current key) | First seen per issuer |
| 2010[a] | 1,392 | 8.6 | 7,835 | 7,835 | 7,835 | 1,389 | 8.6 | 7,819 | 5.6 |
| 2011 | 1,424 | 9.5 | 6,679 | 6,679 | 6,679 | 1,389 | 9.6 | 6,533 | 4.7 |
| 2012 | 1,472 | 9.7 | 6,444 | 6,444 | 6,444 | 1,389 | 9.9 | 6,125 | 4.4 |
| 2013 | 1,539 | 9.5 | 6,286 | 6,385 | 6,286 | 1,389 | 9.7 | 5,602 | 4.0 |
| 2014 | 1,619 | 10.1 | 6,732 | 7,004 | 6,732 | 1,389 | 10.1 | 5,583 | 4.0 |
| 2015 | 1,689 | 10.1 | 6,675 | 7,092 | 6,734 | 1,389 | 10.1 | 5,288 | 3.8 |
| 2016 | 1,744 | 9.4 | 6,240 | 6,710 | 6,372 | 1,389 | 9.4 | 4,810 | 3.5 |
| 2017 | 1,810 | 9.6 | 6,490 | 7,120 | 6,725 | 1,389 | 9.4 | 4,674 | 3.4 |
| 2018 | 1,893 | 9.5 | 6,815 | 7,529 | 7,115 | 1,389 | 9.4 | 4,719 | 3.4 |
| 2019 | 1,973 | 9.5 | 6,837 | 7,644 | 7,208 | 1,389 | 9.3 | 4,496 | 3.2 |
| 2020 | 2,089 | 8.6 | 7,212 | 8,010 | 7,649 | 1,389 | 8.2 | 4,380 | 3.2 |
| 2021 | 2,247 | 9.5 | 8,216 | 9,191 | 8,732 | 1,389 | 8.9 | 4,505 | 3.2 |
| 2022 | 2,291 | 9.9 | 8,158 | 9,135 | 8,689 | 1,389 | 9.3 | 4,447 | 3.2 |
| 2023 | 2,329 | 9.9 | 7,977 | 9,061 | 8,557 | 1,389 | 9.1 | 4,251 | 3.1 |
| 2024 | 2,387 | 9.4 | 7,819 | 8,954 | 8,500 | 1,389 | 8.8 | 4,085 | 2.9 |
| 2025 | 2,453 | 9.5 | 7,990 | 9,252 | 8,719 | 1,389 | 8.8 | 3,926 | 2.8 |
| 2026[b] | 2,498 | 7.6 | 6,694 | 7,778 | 7,342 | 1,389 | 7.1 | 3,192 | 2.3 |

[a]Left-censored year (a project under way before the corpus start is first seen at its first in-window filing); [b]partial year (filings to 2026-10-27). "First seen" counts projects whose earliest mention falls in the year; the windowed keys split a current-key project into episodes whenever the gap between consecutive mentions exceeds $k$ years (131,823 episodes at $k = 3$ and 126,318 at $k = 5$ against 121,099 projects). The fixed cohort holds the issuer composition constant: its mentions per issuer are flat while its new projects per issuer fall, which is consistent with a time-invariant key attaching later mentions to existing projects, with a change in disclosure, or with both; the series do not discriminate between these readings.

**Table 28.** Issuers with the most resolved projects

| Ticker | Issuer | Resolved projects |
|---|---|---|
| UEC | Uranium Energy | 240 |
| UUUU | Energy Fuels | 214 |
| JBLU | Jetblue Airways | 203 |
| HAL | Halliburton | 201 |
| RIOT | Riot Platforms | 195 |
| EHC | Encompass Health | 191 |
| EBS | Emergent Biosolutions | 187 |
| CDE | Coeur Mining | 180 |
| MUX | Mcewen Inc | 180 |
| CLSK | Cleanspark | 170 |
| ALK | Alaska Air Group Inc | 168 |
| CWEN | Clearway Energy Inc Class C | 168 |

Projects under the current resolution key; ties are ordered by ticker. Issuers whose business consists of enumerable sites (mines, wells, routes, data-center sites) accumulate projects by construction of the key, which keys on the first location or name token of each mention (Section 6).

**Table 29.** Self-reported extraction confidence of stored mentions, by run

| Self-reported confidence (rounded) | Mentions | Share (%) | June run (backfilled) | October run |
|---|---|---|---|---|
| 0.5 | 1,611 | 0.5 | 106 | 1,505 |
| 0.6 | 3,278 | 1.1 | 304 | 2,974 |
| 0.7 | 21,647 | 7.0 | 2,239 | 19,408 |
| 0.8 | 72,280 | 23.5 | 8,594 | 63,686 |
| 0.9 | 100,831 | 32.7 | 14,459 | 86,372 |
| 1.0 | 108,418 | 35.2 | 14,228 | 94,190 |
| Total | 308,065 | 100.0 | 39,930 | 268,135 |

Confidence is the value returned by the model in its JSON answer, rounded to one decimal; mentions below 0.45 are discarded before storage. A missing confidence is set to 0.6 by the post-processing code and passes the threshold: 3,278 mentions carry exactly 0.600 and cannot be separated from genuine 0.6 answers because the raw response is not stored. The distribution describes the model's reporting habit, not accuracy; calibration requires the human labels of gate (d) (Table 20).

**Table 30.** Knowledge-graph composition: node and edge types

| | Type | Count |
|---|---|---|
| Edges | `owns` | 121,099 |
| Edges | `located_in` | 54,583 |
| Edges | `uses` | 28,774 |
| Nodes | `Project` | 121,099 |
| Nodes | `Partner` | 36,822 |
| Nodes | `Location` | 14,508 |
| Nodes | `Technology` | 8,551 |
| Nodes | `Company` | 2,492 |
| Nodes | total | 183,472 |
| Edges | total | 204,456 |

Nodes are projected from the resolved projects (one node per company, project, distinct location string, technology tag and partner string); edges are `owns` (company–project), `located_in` and `uses`. Partner nodes are created without edges in the current build. Technology tags are free text and are grounded in the section text for only part of the mentions (Table 17).

**Table 31.** Most frequent name tokens used as resolution keys for location-less mentions

| Name token (key) | Mentions | Projects | Share of token-keyed mentions (%) | Share of token-keyed projects (%) |
|---|---|---|---|---|
| acquisition | 40,407 | 2,256 | 21.4 | 3.4 |
| share | 5,106 | 2,060 | 2.7 | 3.1 |
| expansion | 2,645 | 1,171 | 1.4 | 1.8 |
| stock | 2,624 | 1,146 | 1.4 | 1.7 |
| investment | 2,054 | 1,086 | 1.1 | 1.6 |
| merger | 3,357 | 895 | 1.8 | 1.3 |
| sale | 2,694 | 842 | 1.4 | 1.3 |
| development | 4,514 | 805 | 2.4 | 1.2 |
| strategic | 1,358 | 629 | 0.7 | 0.9 |
| capital | 2,295 | 562 | 1.2 | 0.8 |
| integration | 1,436 | 558 | 0.8 | 0.8 |
| credit | 635 | 473 | 0.3 | 0.7 |
| restructuring | 1,650 | 468 | 0.9 | 0.7 |
| research | 1,160 | 416 | 0.6 | 0.6 |
| product | 1,004 | 386 | 0.5 | 0.6 |

The key for a mention without an extracted location is the first token of at least four letters in the normalised project name (after a short stop-word list). Generic head words therefore merge unrelated initiatives of the same issuer and type into one project; the full list is in `results/resolver/top_name_tokens.csv`.

**Table 32.** Over-merge indicators of the resolution key, by category

| Category | Projects | ≥ 2 distinct names (%) | ≥ 3 distinct names (%) | ≥ 3 distinct first partners (% of projects with a partner) | All names distinct (% of multi-mention) |
|---|---|---|---|---|---|
| M&A / integration | 41,742 | 30.9 | 14.6 | 10.9 | 30.2 |
| Capital expenditure program | 10,720 | 19.4 | 6.4 | 4.1 | 38.9 |
| Strategic partnership | 9,909 | 19.7 | 6.7 | 4.3 | 38.6 |
| Geographic expansion | 7,868 | 25.2 | 10.2 | 1.9 | 35.8 |
| Plant / site construction | 7,590 | 30.3 | 13.5 | 4.0 | 34.7 |
| Cost reduction / restructuring | 6,481 | 17.5 | 5.7 | 1.6 | 26.4 |
| Drug / clinical pipeline | 5,360 | 46.2 | 27.9 | 12.0 | 28.1 |
| New product / launch | 5,356 | 28.0 | 12.2 | 3.4 | 37.9 |
| R&D program | 5,080 | 27.5 | 11.4 | 3.6 | 30.1 |
| Infrastructure investment | 4,481 | 20.8 | 7.5 | 2.2 | 38.1 |
| Manufacturing / capacity expansion | 4,140 | 27.1 | 11.3 | 2.2 | 34.8 |
| Technology deployment | 4,128 | 15.8 | 4.8 | 1.8 | 34.5 |
| Digital transformation | 2,642 | 19.3 | 6.3 | 0.0 | 25.9 |
| ERP / systems implementation | 1,735 | 17.2 | 3.5 | 1.4 | 12.9 |
| Sustainability / ESG | 1,305 | 15.2 | 6.1 | 1.1 | 26.8 |
| Supply-chain initiative | 684 | 15.1 | 3.5 | 1.0 | 27.0 |
| Data center | 449 | 26.1 | 9.4 | 3.6 | 46.5 |
| Energy transition / renewables | 415 | 24.3 | 10.1 | 1.6 | 38.4 |
| AI / ML initiative | 376 | 16.2 | 4.8 | 0.0 | 27.8 |
| Automation / robotics | 374 | 13.1 | 2.7 | 4.2 | 28.0 |
| Cloud migration | 264 | 17.1 | 5.7 | 0.0 | 21.9 |

Distinct names and partners are counted over the mentions of each project (lower-cased); three or more distinct counterparties in one M&A project is an over-merge by construction. For 1,630 of 12,372 M&A projects in which both are named, the project amount comes from a mention naming a different counterparty than the representative mention. These are indicators, not error rates: the linkage audit of gate (d) has not been run.

## G. Stage D audit exhibits

The exhibits generated from the read-only audits of this revision (`code/audit/`, outputs in `results/`) are placed in the sections that discuss them: the ledger and outage reconciliation in Section 5.5 (Tables 7 and 8), the focus-window coverage in Section 5.3 (Table 6 and Figure 3), the financing class in Section 5.1 (Table 5), the specification curve in Section 6.2 (Table 9 and Figure 4), the cohort regression in Section 7.3 (Table 15), the grounding, 8-K consistency and rules-baseline results in Section 8.2 (Tables 17 to 19 and Figure 8), and the universe artefacts, cohort reconciliation, time series, name tokens and over-merge indicators in Appendices E and F.

## H. Replication

1. Clone the repository at revision `2324b58` and create the Python environment from the lock files shipped with the paper (`.python-version` = 3.12.15; `requirements.lock`, 88 packages

from the master node, including DuckDB 2.0.0, pyarrow, lxml and sentence-transformers; `environment.md` records the hardware). The exhibit environment of the paper is pinned separately in `reproducibility/environment/`.

2. `python scripts/build_universe.py` rebuilds `data/metadata/universe.json` from a *current* download of the holdings and ticker files, which will not reproduce the October 2026 snapshot; the frozen artefacts used here are kept with the paper (`data/metadata/`, SHA-256 in `SHA256SUMS`) and must be used instead.
3. `python scripts/download_filings.py --universe --rate 9` downloads the corpus (idempotent; about 254.5 GB; seven hours at two IP addresses in October 2026). The cluster orchestration used for the October 2026 build is in `scripts/cluster/`.
4. `python scripts/build_database.py`, `parse_filings.py`, `build_embeddings.py`, `build_factors.py` build the corpus layer; `export_shard.py`/`merge_shards.py` are needed only for a distributed build. The S&P 500 10-K items of the June 2026 block cannot be rebuilt by this path (Section 4.3).
5. In `project-db/`: `python -m spid pending` reports the sections to be extracted and the estimated cost; `python run_full_extract.py --workers 48` runs the extraction against the ledger (a re-run is not expected to return identical mentions; Section 5.4); `python -m spid resolve graph embed analytics` builds the project layer (representative fields depend on tie order; Section 6.1).
6. In the working-paper folder: `./run_all.sh` regenerates every table, figure and inline number from `data/` and `results/` and runs the 22 checks of `tests/`; `./run_all.sh --with-node` first re-runs `code/export_stats.py` read-only on the master node, and `--with-audit` re-runs the Stage D audits of `code/audit/` (read-only connections, temporary tables only, seed 20261105). `latexmk` in `paper/` compiles the manuscript.