Contains EDA Separately Analyzed from the notebook - eda_experiments.ipynb in the notebooks folder.

Doc Owner - Joel Markapudi.

Date - 05/10/2024.

Starting Analysis

First View Analysis On Content:

Three columns contain nested JSON/dict structures:

```
Column          Structure                               Content
labels          {'1d': 0, '30d': 0, '5d': 1}        Binary classification targets (market
movement?)
returns         Nested dict with 1d/5d/30d keys     Stock price data: closePriceStartDate,
closePriceEndDate,
                                                    ret (return %), date
tickers         [AIR]                               List of ticker symbols
```

1. These are ML targets. Cannot directly store these in Qdrant payload (need flattening)
2. Tickers list: Understand this better, soon.

```
Row 0: "ITEM 1.BUSINESS General AAR CORP. and its subsidiaries..." (180 chars)
Row 1: "AAR was founded in 1951, organized in 1955..." (84 chars)
Row 2: "We are a diversified provider of products..." (121 chars)
```

1. Sentences are coherent, complete thoughts.

```
"ITEM 1.BUSINESS General AAR CORP. and its subsidiaries are referred to..."
 └───────────────┘
   This part
```

- The text itself starts with "ITEM 1.BUSINESS" as part of the sentence content. It's not in a separate column - it's embedded in the sentence text.
- Are cik, section, reportDate actually useful for filtering? (Do they have good distribution?)
- Are there other patterns we haven't discovered yet?

From Next cells:

- Expected: 10 sections (0-9 for 10-K items)
- Actual: section has 20 unique values
  - What this means:
  - Either sections go beyond 0-9 (10-19 range?)
  - OR there are sub-sections (e.g., 1.1, 1.2 encoded as 11, 12?)

```
Field        Unique Values          Interpretation                      Filtering Value
cik              10                     10 companies only                   ✅ HIGH - company
filter
name         10                 1:1 with CIK                    ✅ HIGH - display name
tickers      10                 1:1 with company                    ✅ HIGH - user-friendly
filter
section      20                 ⚠️ MORE than expected             ✅ HIGH - but needs mapping
docID        188                ~19 docs/company                    ✅ MEDIUM - document-level
filter
filingDate       181                    Nearly 1:1 with docID            ⚠️ LOW - use
reportDate instead
reportDate       91                     ~2 filings/period                   ✅ MEDIUM - time
filter
sentence         96,465             48% unique                          ❌ N/A - this is the
content
sentenceID       200,000            100% unique                         ✅ HIGH - primary key
```

- 188 documents total across 10 companies:
  - Average: 18.8 documents per company
  - This suggests ~19 years of filings (if annual 10-Ks)
- 91 unique reportDates vs 181 filingDates:

        - Multiple companies share same fiscal year-ends
        - Common fiscal year-ends: Dec 31, Jun 30, Sep 30
        - filingDate spread (181 unique) = different filing times
- Three binary sentiment labels (1d, 5d, 30d) derived from market reaction windows.
- Potential supervised target for finetuning sentiment or volatility predictors.
- Label-conditioned embeddings: correlate language tone with short-term market moves.
- Later: contrastive training of “risk-positive vs risk-negative” sentences.

Metadata Richness:

- can enrich embeddings with these categorical tags (via adapters or metadata vectors).
- these features are extremely useful for metadata-filtered search and bias analysis later.

Text Content Insights

- 96,465 unique sentences out of 200,000 total:
    - Duplication rate: 51.7% (103,535 repeated sentences)
    - This is NORMAL for 10-Ks because:
        - Boilerplate language (risk disclaimers, accounting policies)
        - Repeated across years ("We are incorporated in Delaware...")
        - Standard regulatory phrases

For RAG:

- Duplicates are FINE (same sentence, different context/year)
- Embeddings will cluster similar content

Low-value fields (don't change, not useful for filtering):

```
Field                   Unique      Why Low Value
entityType              1           Always "operating" (no variance)
tickerCount             1           Always 1 ticker (derived field)
form                    1           Always "10-K" (dataset definition)
```

```
exchanges               3           Only 3 exchanges (NYSE, NASDAQ, ?) - low filtering value
stateOfIncorporation    5           Only 5 states (mostly DE) - not relevant for financial
analysis
```

Medium-value field:

```
Field                   Unique      Potential Uses
sic10                   10          Industry codes (1:1 with company, but useful for "show me
tech companies")
```

ML Target Fields (Finance-Specific)

- labels structure: {'1d': 0, '30d': 0, '5d': 1}
- Binary classification: Did stock go UP (1) or DOWN/FLAT (0)?
- Timeframes: 1-day, 5-day, 30-day post-filing
- 8 unique combinations = all possible 0/1 patterns ($2^3 = 8$)
- returns structure: Nested price data
- 188 unique values = 1 per document (document-level targets, not sentence-level)
- Contains: start/end prices, return %, dates
- Used for regression tasks (predict magnitude of movement)

Temporal Metadata Analysis

Three timestamp fields:

```
Field              Unique      Granularity        Use Case
reportDate         91          Fiscal period end  ✅ "Show me Q4 2019 results"
filingDate         181         SEC filing date    ⚠️ "When did market learn this?"
acceptanceDateTime 188         Exact timestamp    ❌ Too granular (hour/minute irrelevant)
```

- Why reportDate (91) < filingDate (181)?
- Multiple companies file on same calendar date (e.g., 2020-02-28)
- But their fiscal year-ends differ (some Dec 31, some Jan 31, some custom)
- ❌ labels, returns (ML targets)
- ❌ entityType, tickerCount, form (no variance)

- ❌ exchanges, stateOfIncorporation (low value)
- ❌ acceptanceDateTime (too granular)
- ❌ sentenceCount (internal counter)

Chunking:

- Chunking window ≈ 3–5 sentences will yield ~100–150 tokens — perfect for encoder models (MiniLM, E5, Titan Embeddings).
- Avoid over-chunking: 10-K prose is repetitive; smaller chunks improve recall for factual retrieval.
- "sentenceID" includes hierarchy: cik_form_year_section_index — perfect for traceability and citation.

Consideration	Insight from EDA	Recommended Approach
Chunk length	Sentences average 28 words; coherent across 2–5-sentence spans.	Use sliding window of 3 sentences (≈ 100–150 tokens).
Chunk boundaries	Section IDs (0–19) define strong topical boundaries.	Chunk within each section; reset window at new section.
Metadata filters	CIK, section, reportDate are perfectly populated.	Use these as metadata filters in vector DB (OpenSearch).
Embedding schema	Text + metadata + docID	Each vector record → `{cik, section, reportDate, sentence_text, embedding}`.
Edge cases	Section imbalance (see dataset card: Item 7 > Item 14 etc.)	Weighted sampling or per-section retrieval balancing.

Part 2, 3 : Distribution Analysis & Sections. And, Text Density Analysis. etc.

EDA Deep Analysis - Part 1: Company, Temporal & Section Analysis

1. COMPANY DISTRIBUTION (Q2) - SEVERE IMBALANCE

Table 3 Summary:

Company	Ticker	SIC	Sentences	Filings	Date Range	% of Total
ADVANCED MICRO DEVICES INC	N/A	3674	38,799	24	1993-2020	19.4%
ABBOTT LABORATORIES	N/A	2834	30,554	25	1993-2020	15.3%
Air Products & Chemicals	N/A	2810	26,282	20	2001-2020	13.1%
CECO ENVIRONMENTAL CORP	N/A	3564	24,867	17	2004-2020	12.4%
AAR CORP	N/A	3720	20,350	21	1994-2020	10.2%
BK Technologies Corp	N/A	3663	19,081	21	1995-2020	9.5%
ACME UNITED CORP	N/A	3420	15,849	26	1995-2020	7.9%
ADAMS RESOURCES	N/A	5172	14,964	19	2002-2020	7.5%
WORLDS INC	N/A	7372	7,797	13	2008-2020	3.9%
Matson, Inc.	N/A	4400	1,457	2	2019-2020	0.7%

Key Stats:

- **Imbalance ratio: 26.63x** (AMD: 38,799 vs Matson: 1,457)
- Std deviation: 10,874 sentences
- Industry diversity: semiconductors, pharma, chemicals, aerospace, energy, shipping

Impact on RAG:

- Retrieval bias toward AMD (20x more chunks than Matson)
- Matson effectively invisible without weighting
- **Solution needed:** Per-company retrieval quotas OR stratified sampling

---

2. TEMPORAL DISTRIBUTION (Q3) - RECENCY BIAS

Coverage Timeline:

Period	Sentences/Year	Filings/Year	Phase
1993-2001	370-1,510	1-3	Sparse (5% of data)
2002 (inflection)	6,361	7	Major jump
2002-2020	9,000-12,600	8-10	Stable (95% of data)

Key Findings:

- Total span: 28 years (1993-2020)
- Usable data: 18 years (2002-2020 only)
- **1999 anomaly:** Only 370 sentences (data gap)
- **2020 peak:** 12,595 sentences (COVID disclosures)
- **NOT evenly spread** - 95% concentration post-2002

Recommendation: Filter `reportDate >= "2002-01-01"` for reliable temporal analysis

---

3. SECTION CODES (Q1) - 20 SECTIONS DECODED

Major Sections:

Section	10-K Item	Sentences	%	Avg Tokens	Status
10	**Notes to Financials**	**60,256**	**30.1%**	26.2	**CRITICAL**
8	MD&A	47,677	23.8%	26.0	High value
1	Risk Factors	24,627	12.3%	27.7	High value
0	Business	21,311	10.7%	25.4	High value
19	Exhibits	14,312	7.2%	28.4	Boilerplate
4	Legal Proceedings	4,534	2.3%	23.6	Standard
9	Financial Statements	3,993	2.0%	22.7	Tables
5	Mine Safety	3,893	1.9%	11.6	Standard

Section	10-K Item	Sentences	%	Avg Tokens	Status
6	Market for Stock	2,836	1.4%	23.5	Standard
3	Properties	2,317	1.2%	20.7	Standard
2	Unresolved Comments	374	0.2%	**4.5**	**NOISE**
7	Reserved	1,355	0.7%	20.5	**SPARSE**
11	Market Risk	608	0.3%	**10.3**	**NOISE**
13	Unknown	479	0.2%	**4.7**	**NOISE**

Extended Sections (11-19): Controls, certifications, exhibits - mostly < 1% each

Critical Insights:

- **Section 10 (Notes) is THE priority** for KPI context (30% of all data)
- Sections 0, 1, 8, 10 = **85% of data** (focus here for RAG)
- Sections 2, 7, 11, 13, 17 = **NOISE** (< 1%, fragment sentences like "See Exhibit 10.1")
- Section 19 (Exhibits) = 7% but legal lists (low semantic value)

Section Code Mapping:

```
CORE (0-9):
0  → Item 1: Business
1  → Item 1A: Risk Factors
2  → Item 1B: Unresolved Staff Comments (SPARSE)
3  → Item 2: Properties
4  → Item 3: Legal Proceedings
5  → Item 4: Mine Safety
6  → Item 5: Market for Stock
7  → Item 6: Reserved (EMPTY)
8  → Item 7: MD&A
9  → Item 8: Financial Statements

EXTENDED (10-19):
10 → Notes to Financial Statements (DOMINANT)
11 → Quantitative Market Risk
```

```
12 → Controls & Procedures
13 → Unknown (SPARSE)
14 → Principal Accountant Fees
15 → Exhibits Index
16 → Form 10-K Summary
17 → Unknown (SPARSE)
18 → Unknown
19 → Exhibit Documents
```

ANSWERS TO 3 OPEN QUESTIONS

Q1: What are the 20 section codes?
Sections 0-9 = standard 10-K items. **Section 10 = Notes to Financial Statements (30% of data - THE KEY SECTION for KPI context).** Sections 11-19 = extended disclosures (exhibits, certifications) - mostly noise.

Q2: Are companies evenly distributed?
NO. Severe imbalance: 26.63x ratio (AMD 19.4% vs Matson 0.7%). Must implement per-company retrieval quotas or stratified sampling to prevent AMD bias.

Q3: Are filings evenly spread over time?
NO. Heavy recency bias: 95% of data is post-2002. Pre-2002 period (1993-2001) is sparse and unreliable for trend analysis.

4. TEXT DENSITY & CHUNK SIZE VALIDATION

Overall Token Statistics

Metric	Value	Implication
Mean	25.8 tokens/sentence	Typical sentence length
Median	22 tokens	Normal distribution (not skewed)
P95	55 tokens	Outliers start beyond this
Max	737 tokens	Tables-as-text (toxic)

Chunk Size Validation

```
3-sentence chunks → ~77 tokens   ✅ SAFE (well under 512 limit)
5-sentence chunks → ~129 tokens  ✅ SAFE (comfortable margin)
19 sentences max  → ~512 tokens  ⚠️ Theoretical max (not recommended)
```

Recommendation: 3-sentence sliding window with 1-sentence stride

- Average: 77 tokens/chunk
- Overlap: 2 sentences preserved (context continuity)
- Output: ~200k chunks from 200k sentences
- Rationale: Prevents topic drift (financial text jumps topics frequently)

Section-Specific Density Analysis

Table 6: Text Density by Section

Section	Item	Avg Tokens	Median	Max	Quality Rating
12	Controls	**33.4**	28	179	Densest (regulatory)
19	Exhibits	28.4	22	672	Dense but boilerplate
1	Risks	27.7	24	362	Good semantic content
10	Notes	26.2	23	428	**IDEAL for RAG**
8	MD&A	26.0	23	433	**IDEAL for RAG**
0	Business	25.4	21	737	Good semantic content
5	Mine Safety	**11.6**	10	101	Sparse
11	Market Risk	**10.3**	2	113	**SPARSE/NOISE**
2	Unresolved	**4.5**	2	55	**FRAGMENT SENTENCES**
13	Unknown	**4.7**	3	74	**FRAGMENT SENTENCES**

Sections Good for RAG: 0, 1, 8, 10 (25-28 tokens, coherent narratives)
Sections Bad for RAG: 2, 7, 11, 13 (4-10 tokens, fragments like "San Francisco, CA")

---

5. OUTLIER ANALYSIS - DATA QUALITY FLAGS

Extreme Outliers (>1000 chars)

Section	Chars	Content Type	Example Preview
1 (Risks)	1,174	Legal disclaimers	"Consequently, we are subject to military conflicts, civil..."
10 (Notes)	1,022	**Financial table as text**	"Sales by segment for these customers are as follows: AAR CORP..."
12 (Controls)	1,040	Regulatory boilerplate	"The Company's internal control over financial reporting is a process..."
19 (Exhibits)	1,330	**Exhibit list**	"4.3 Description of Capital Stock (filed herewith) 4.4 Rights Agreement..."
19 (Exhibits)	1,850	**Material contracts list**	"Material Contracts 10.1* Amended and Restated AAR CORP. Stock Benefit..."

Problem: These break embeddings (737-token max observed → truncation) and have no semantic value

Solution: Filter sentences > 500 chars (keeps P95+ data, removes 2% toxic outliers)

ACTIONABLE DECISIONS FOR RAG PIPELINE

Decision 1: Filtering Strategy

```
df_clean = df.filter(
    # Remove noise sections
    ~pl.col("section").is_in([2, 7, 11, 13, 17]) &

    # Remove outliers (tables-as-text, exhibit lists)
    (pl.col("sentence").str.len_chars() <= 500) &

    # Remove sparse temporal data
    (pl.col("reportDate") >= "2002-01-01")
```

```
)

# Expected result: ~180k sentences (removes 10% noise, keeps 90% quality data)
```

Impact:

- Removes 2, 7, 11, 13, 17 (< 2.5% of data, fragments)
- Removes outliers > 500 chars (~2% of data, tables/lists)
- Removes pre-2002 data (~5% of data, sparse coverage)
- **Total removed: ~10% | Quality retained: ~90%**

Decision 2: Chunking Strategy

Recommended: 3-sentence sliding window, 1-sentence stride

Rationale:

- 77 tokens avg (safe for 512-token models)
- Preserves context via overlap
- Prevents topic drift (financial text jumps topics frequently: KPI → explanation → next KPI)
- Shorter chunks = better precision for KPI extraction

Alternative considered: 5-sentence chunks (129 tokens)

- Rejected because: longer chunks risk topic drift within chunk

---

Decision 3: Company Balancing

Problem: 26.63x imbalance means AMD dominates retrieval

Options:

Strategy	Pros	Cons
A. Downsample AMD/Abbott	Balanced training	Loses information

Strategy	Pros	Cons
B. Weighted retrieval	Keeps all data	Complex implementation
C. Per-company quotas	Guarantees diversity	May miss best match

Recommendation: Option C (Per-company retrieval quotas)

- Retrieve top-3 results per company
- Then rank all 30 results by similarity
- **Why:** FinSight KPI extraction benefits from company diversity (prevents "AMD-only" responses)

Decision 4: Priority Sections for RAG

Focus on these sections (85% of data):

1. **Section 10 (30%)** - Notes to Financial Statements → KPI context, explanations
2. **Section 8 (24%)** - MD&A → Narrative analysis, trends
3. **Section 1 (12%)** - Risk Factors → Qualitative insights
4. **Section 0 (11%)** - Business → Company overview, revenue streams

Optionally include:

- Section 4 (2.3%) - Legal Proceedings (if relevant)
- Section 9 (2.0%) - Financial Statements (tables - handle carefully)

Exclude:

- Sections 2, 7, 11, 13, 17 (noise)
- Section 19 (7%) - Exhibits (boilerplate lists)

METADATA FOR QDRANT PAYLOAD - potential schema

Based on analysis, recommended payload schema:

```
{
    "chunk_id": "0000001750_10-K_2020_section_8_chunk_42",
    "text": "[3-sentence chunk text]",
    "cik": "0000001750",
    "company": "AAR CORP",
    "ticker": "AIR",
    "section": 8,                  # Section code (0-19)
    "reportDate": "2020-05-31",
    "docID": "0000001750_10-K_2020",
    "sic": "3720"                  # Industry code (optional)
}
```

Filterable fields: `cik`, `section`, `reportDate`, `ticker`

Stored but not indexed: `docID`, `sic`, `company`

SUMMARY: KEY TAKEAWAYS

Data Characteristics

- 200k sentences → **~180k usable** (after filtering)
- 10 companies, **severe imbalance** (26.63x)
- 28-year span, **95% post-2002** (recency bias)
- 20 sections, **4 sections = 85% of data** (0, 1, 8, 10)

Text Properties

- Average: 25.8 tokens/sentence
- 3-sentence chunks: 77 tokens (safe for embeddings)
- Outliers: 2% of data (tables-as-text, exhibit lists)

Critical Sections

- **Section 10 (30%)**: THE priority for KPI context
- Sections 0, 1, 8: High-value narrative content
- Sections 2, 7, 11, 13: Noise (filter out)

Required Actions

1. Filter noise sections (2, 7, 11, 13, 17)
2. Remove outliers (> 500 chars)
3. Use 3-sentence sliding window chunking
4. Implement per-company retrieval quotas
5. Filter reportDate >= 2002-01-01

Next Steps

- Proceed to Section 1.4 (if needed): N-gram analysis, vocabulary patterns ?? Think about this. This is small_full.

Deep-dive insights from EDA (Q2 → end)

1) Company distribution (Q2)

What you found: 10 companies, **200,000 sentences** total; **large imbalance** (e.g., AMD ≈ 38.8k sentences vs Matson ≈ 1.5k). **Imbalance ratio ~26.6×**; filings per company vary (2 → 26) and span **1993–2020**.

Why this matters

- **Index skew**: a few firms dominate the vector index. Pure ANN retrieval may bias toward overrepresented writing styles/phrases.
- **Evaluation skew**: if your gold set concentrates in "big" companies/years, you'll overestimate performance.

Actions

- **Balanced gold set**: sample 2–3 filings per company across early/mid/late years (e.g., 2000, 2010, 2019) → fair coverage.
- **Index caps**: per company, cap max vectors per section/year (or down-weight when ranking).
- **Stratified eval**: report metrics per-company and macro-average across companies so small issuers don't get hidden.

---

2) Temporal distribution by year (Q3)

What you found: coverage **1993–2020**, steady growth post-2002 (SOX era) and again in late 2010s. Sentences/year ~7.1k on avg, peaks around 2020.

Why this matters

- **Language drift**: disclosure tone, accounting phrasing, and risk taxonomy evolved.
- **Section composition shift**: some items (e.g., MD&A, controls, exhibits) grew over time.

Actions

- **Decade shards**: (optional) build decade/tag filters to study retrieval drift (90s/00s/10s).
- **Recency weighting**: for live use, prefer latest year passages when period isn't explicit.
- **Generalization check**: train prompts/heuristics on pre-2015 filings, validate on 2016–2020; watch drops.

---

3) Section code distribution (Q1) & cross-company heatmap

What you found: 20 section codes (0–19), not just 0–9. Heavy hitters: **10 (30.1%)**, **8 (23.8%)**, **1 (12.3%)**, **0 (10.7%)**, **19 (7.2%)**. Very light: **2, 11, 13, 15, 17**. Heatmap confirms most companies populate the heavy sections; some sparsity in others.

Interpretation (practical)

- The dataset collapses more than the canonical 10 items—likely sub-items/appendices are mapped to higher codes (10–19).
- High-volume sections (8/10/1/0) drive most of your retrieval hits; thin sections will hurt recall if you rely on them.

Retrieval priors (policy)

- **KPI extraction** → bias to **8, 10** (financial statements & notes) and **7/MD&A-like** areas if present.
- **Risk/Drivers** → bias to **1A/7-like** codes (your heavy **1, 0** buckets often carry business/MD&A-type prose).
- Keep a **down-weight** for **19** (exhibits/references) unless you specifically need exhibits.

(Later, we can learn a compact mapping "code → canonical item label" by sampling top n-grams per code.)

---

4) Token length distribution & chunking

What you found: Mean ~25.8 tokens/sentence, **p95 ~55**, **max ~737** (tables/lists). Your table of density by section shows **avg tokens** vary widely (**section 12 ~33.4** densest; **section 2 ~4.5** sparsest).

Decisions

- **Adaptive chunking** (per section density):
  - **Dense sections (avg ≥ ~26 tokens) → 3-sentence window**, **1-sentence overlap**.
  - **Medium (18–26) → 4-sentence window**, **1-sentence overlap**.
  - **Sparse (≤ ~18) → 5–6 sentences**, **2-sentence overlap**.
- **Hard caps**: truncate chunks at **~150–200 tokens** (keeps encoders efficient; nice fit for rerankers too).
- **Reset on section change** to avoid cross-topic chunks.

Outlier handling

- Sentences **>1000 chars** are often lists/tables/exhibits; treat as **table-like**.
  - If KPI-targeted, run a **regex/table parser** path; else **exclude from text embeddings** to reduce noise.
  - Tag these in metadata (`is_table_like=1`) for optional specialized handling.

---

5) Duplicates & boilerplate

What you observed in your notes: ~**52%** of sentences are duplicates (not surprising): boilerplate risk/legal text, multi-year carry-overs.

Why this helps (if managed)

- Embeddings of duplicated boilerplate will cluster; ANN can over-return them.

Actions

- **Near-duplicate suppression at index time**: within each `(cik, section, decade)` drop vectors with cosine sim ≥ **0.97** to an existing exemplar.
- **Query-time down-weight** duplicates (feature `dup_count`) so unique, data-rich passages rank higher.

---

6) Metadata you can reliably filter on

- **High value**: `cik`, `reportDate`, `section`, `docID` (audit trail), `sentenceID` (citation), `sic` (industry).
- **Low value**: `entityType`, `tickerCount`, `form` (constant); `stateOfIncorporation`, `exchanges` (coarse, rarely useful).
- **Targets**: `labels`, `returns` at **document** level (not sentence level) — useful for downstream supervised tasks, not for RAG retrieval directly.

OpenSearch mapping sketch

- `text`: the chunk text
- `vector`: dense embedding
- `cik` (keyword), `reportDate` (date), `section` (short), `docID` (keyword), `sentence_span` (short), `sic` (keyword)
- `char_len`, `token_len`, `is_table_like`, `dup_count` (ints) for ranking rules

---

7) Retrieval strategy that fits these distributions

1. **Constrain early with metadata**: `(cik, reportDate)` (if user specified), then **section priors** by intent.
2. **Hybrid search**: vector ANN + **keyword filters** ("in millions", "Net sales", KPI labels) improves precision in dense sections.
3. **Rerank small k** (optional later): a cross-encoder reranker (or Bedrock "judge" prompt) on top-30 → top-5 improves faithfulness.
4. **Evidence guardrails**: only accept KPI if **evidence sentence contains the number & scale tokens** (prevents "off-by-scale" errors).

---

8) KPI extraction implications

- Most KPI sentences will live in **8/10**; narrative drivers in **1/0/7-like**.
- Your **unit normalization** must handle "in millions/billions" headers; add a **page/paragraph-level scope detector** (regex on a few neighboring chunks).
- **Period alignment**: prefer `reportDate` for fiscal tagging; if period text is ambiguous in MD&A, fall back to the **nearest financial-statement chunk** for the same metric.

---

9) Evaluation slices to add (so results are credible)

Report all metrics **by**:

- **Company size**: top-3 vs bottom-3 by sentence_count
- **Year bucket**: pre-2005, 2005–2014, 2015–2020
- **Section group**: {8/10}, {1/0/7}, {others}

This guards against a system that looks good only on AMD-style heavy disclosures or only on recent years.

---

10) Concrete next steps (fast to execute)

1. **Build a section-aware chunker** with the adaptive window rules above (store `token_len`, `is_table_like`).

2. **Index with duplicate suppression** (cos ≥ 0.97 within `(cik, section, decade)`).

3. **Write retrieval policies** (KPI vs Narrative) with section priors and a few keyword hints per KPI (e.g., Revenue, Net income, R&D, Operating income).

4. **Assemble a balanced gold set** (2–3 filings × 10 companies, spread across years) and lock the schema for scoring.

5. **Run a small ablation**:

   - fixed 3-sent chunks vs adaptive chunking
   - vector-only vs hybrid+keyword filters
   - with vs without duplicate suppression → Pick the combo that maximizes Recall@10 and KPI EM on the gold set.

TL;DR design decisions you can lock now

- **Adaptive chunking** by section density; **reset at section boundaries**; cap at ~200 tokens; 1–2 sentence overlap.
- **Index controls**: metadata filters; duplicate suppression; flag `is_table_like`.
- **Retrieval priors**: KPI → {8,10}; Narrative → {1,0,7}; down-weight {19} unless needed.
- **Evaluation**: stratify by company/period/section group; balance the gold set.
- **Normalization**: enforce evidence-contains-number rule; handle “in millions/billions” via neighborhood regex.

Deep Analysis Addendum (Essential Highlights Only)

This addendum captures **new, actionable** findings from the auxiliary analysis that complement our main EDA brief. It excludes items already covered or proven incorrect.

A) KPI Signal Density — Where Numbers Actually Live

Why this matters: Directly informs **section prioritization** for structured KPI extraction (numbers, units, EPS, YoY) vs. narrative-only RAG.

Key takeaways (consistent with our earlier EDA, now reinforced):

- **Item 8 (MD&A)** — highest overall KPI signal (currency %, growth verbs, some YoY): prime target for *numbers-with-explanations*.
- **Item 10 (Financial Statements/Notes)** — strongest **EPS** & **units ("in millions/billions")** signal: prime target for *audited KPI lines*.
- **Item 7 ("Selected Financial Data" legacy / financials summary)** — surprisingly high **currency** and **units** despite being smaller: treat as **secondary KPI** source.
- **Item 1 (Risk Factors)** — **narrative-rich** (growth verbs) but **number-poor**: keep for *explanations*, not for extraction.

Practical policy (KPI first-pass):

- **Extract** from: **8 (MD&A)** → **10 (Notes)** → **7 (Selected/Financials)**.
- **Explain** from: **1 (Risks)** → **0 (Business)**.

> This validates our **"KPI first, Narrative second"** routing and helps tune budgets (more LLM parsing time where numeric density is high).

B) Section Mapping (Cleaned)

Use the **n-gram signatures + manual verification** to finalize a **human label** per section (only the parts that differed or sharpened our mapping):

Section	Human Label (for UI + Routing)	Notes (why)
0	**Business / Overview**	Terms: products, sales, company, operations
1	**Risk Factors**	Modal verbs (“may”, “could”), “risks”
2	**Unresolved Staff Comments**	“item 1b”, “unresolved staff”, “comments none” → **boilerplate**
7	**Selected Financial Data** (legacy)	“selected financial”, currency/units spikes
8	**MD&A**	“million”, “sales”, “tax”, “income”, “cash”
9	**Financial Statements**	Statements body (narrative around line items)
10	**Notes to Financial Statements**	“financial”, “consolidated”, “december”, “value”, “stock”
11	**Acct. Disagreements**	“disagreements with accountants”, typically **none**
12	**Controls & Procedures**	“internal control”, “over financial reporting”
19	**Exhibits & References**	“form”, “filed”, “report”, index-like cues

Policy impact:

- Treat **2, 11** as **boilerplate / low-value** for KPI; keep searchable for compliance queries.
- Bias KPI retrieval toward **8, 10, 7**; bias explanatory retrieval toward **1, 0**.

C) “Noise” Sections to Down-weight or Filter (KPI Path)

Based on KPI-zero signals and boilerplate cues, **down-weight** (or **skip** for structured extraction) the following:

- **2 – Unresolved Staff Comments** (compliance boilerplate)
- **5 – (As surfaced: low/zero KPI signal in sample)**
- **11 – Disagreements with Accountants** (typically “none”)

- **13 – (As surfaced: negligible KPI content in sample)**

  > Keep these **searchable** for niche questions, but do **not** spend LLM KPI budget here.

D) Section-Aware Chunk Size Defaults (Sharper)

Use KPI density to guide default chunk size:

- **KPI-dense (7, 8, 10): 2–3 sentences** (precise spans; avoid diluting with narrative)
- **Narrative-heavy (0, 1): 4–5 sentences** (context matters; still cap ~200 tokens)

Always **reset at section boundaries** and **flag table-like outliers** (long lists/tables) for specialized handling.

E) Query Routing Patterns (Refined Cheatsheet)

Minimal, high-signal routing based on section labels and n-gram cues:

- **KPI intents** → boost **8, 10, 7**
  Regex hints: `revenue|net income|operating income|EPS|gross margin|R&D|cash|capex|tax`
- **Risk/Qualitative intents** → boost **1, 0**
  Hints: `risk|threat|challenge|uncertainty|supply chain|macro`
- **Controls/Compliance** → boost **12, 11, 19**
  Hints: `internal control|disclosure|procedure|exhibit|agreement`

  > Combine **metadata filters** (`cik` , `reportDate`) with **section boosts** for first-pass retrieval.

F) Important Correction — Duplication Estimation

Do not rely on the reported duplication rates where `n_near_dupes` >> `n_sampled` (impossible).
Likely issues: over-counting cluster pairs, bucket collisions, or cross-section contamination.

What to keep:

- The **directional** reminder that **Risk Factors** and **Controls** carry more boilerplate;

- The **principle** to **apply near-duplicate suppression** (cosine or SimHash) **within** `(docID, section, decade)`.

What to fix later:

- Recompute with **unique cluster counting** (e.g., LSH clusters → count `cluster_size - 1` once),
- Or run **cosine-based suppression** on embeddings directly during index build.

Deep-Dive EDA Briefing (for SEC 10-K sentence dataset)

1) What each artifact tells us (and why it matters)

A. `top_ngrams_by_section.csv` — Section "language fingerprint"

- You computed TF-IDF top n-grams per section, sampled per section, (1,2)-grams with sensible DF thresholds. This gives a *signature vocabulary* for each section (e.g., Item 1: "business", "segment", "customers"; Item 1A: "risk", "adverse"; Item 7: "management discussion", "operations", etc.).

- **Why it matters**:

  - Improves retrieval by adding a prior: given a user intent (e.g., "risks of supply chain"), boost sections whose n-grams match the intent.
  - Enables **section-aware chunking** and **router prompts** (see §3).

B. `section_label_suggestions.csv` — Human-readable section mapping

- You mapped those n-gram signatures to readable labels (Business/Overview, Risk Factors, MD&A, Financial Statements/Notes, Controls & Procedures, Legal/Exhibits). This is exactly the bridge from opaque numeric `section` codes (0–19) to practical filters in UX and routing.

- **Why it matters**:

  - Gives you a clean **taxonomy** to anchor UI filters, metadata filters in retrieval, and evaluation slices.

C. `duplication_by_section.csv` — Near-duplicate pressure by section

- Using a SimHash-style approach (shingles→64-bit hash→prefix buckets→sampled pair checks), you estimated near-duplication rates per section. Given the overall dataset has ~48% unique sentences (duplication is normal in 10-Ks), this tells you where boilerplate repeats the most.

- **Why it matters**:

  - Guides **index compaction** (dedupe or down-weight duplicates inside the ANN index).
  - Helps **evidence diversity**: when forming a context window, avoid stuffing multiple near-duplicates—use one with strongest metadata match.

D. `kpi_signal_scan_by_section.csv` — Where the numbers live

- Regex probes for currency, percent, EPS, units (thousands/millions/billions), YoY/growth verbs, by section with per-section sampling. It tells you *where structured KPIs are likely extractable* (e.g., high numeric density in Financial Statements/Notes and MD&A; lower in Legal/Exhibits).

- **Why it matters**:

  - Narrows **extractor scope** (prioritize sections with high numeric signal).
  - Drives **prompt specialization** (use a KPI template only when signal ≥ threshold).

---

2) Cross-checks from your notebook (foundation facts)

- **Scale/shape**: 200,000 rows × 19 cols; ~144 MB in memory for the small_full parquet in Polars.
- **Time span**: Coverage ~1993–2020 (28 years). Useful for period filters & drift checks.
- **Section codes**: 20 unique (0–19), not just 0–9. Some are exhibits/controls; mapping via label suggestions is needed for UX and routing.
- **Token lengths**: Mean ≈26 tokens/sentence; p95 ≈55; long tails often tables/lists (outliers >1000 chars) in items like 10, 12, 19. Chunking should treat **table-like spans** differently (capture intact or skip).
- **Company imbalance**: Sentence volume per company is imbalanced (expected with 28-year span). Retrieval should **favor doc/time filters** to avoid over-representing prolific issuers.

---

3) Design decisions this EDA unlocks (actionable)

3.1 Chunking & indexing

- **Unit of chunk**: start with **3–5 sentences** (≈ 75–130 tokens avg), which sits well under most 512-token embedding limits and captures local context for KPI lines plus the immediate explanation. Your token stats support this.
- **Table-like outliers**: Detect via `char_count > 1000` (your rule); either (a) capture as **verbatim block** in a separate “table” index with table-aware embedding, or (b) **skip** them for narrative retrieval and rely on structured extraction sourced from those sections when needed.
- **Metadata keys** (store with each vector): `cik`, `name`, `docID`, `section`, `reportDate`, `year`, and your **human section label** from `section_label_suggestions.csv`. These power facet filters and UX pivots.

3.2 Retrieval & routing

- **Intent → Section boost**: Map user intents (e.g., “risk” / “MD&A outlook” / “revenue growth”) to section labels using `top_ngrams_by_section.csv` signatures. Apply a **pre-filter or boost** at retrieval time (metadata filter + query rewrite with section terms).

- **De-dup policy**:

  - **Index-time**: if `dup_rate` is high for a section, keep only one vector per near-duplicate cluster per docID (or store all but mark duplicates with a lower weight).
  - **Query-time**: apply a **diversity constraint**: no two contexts with Hamming distance ≤ T (or same sentence hash) in the final top-k.
- **Temporal filter**: Default to `reportDate` window for comparability (e.g., “show last 3 years”), with optional `filingDate` when event-time matters (market reaction labels are keyed to filing).

3.3 KPI extractor scope

- Use `kpi_signal_scan_by_section.csv` to prioritize **Financial Statements/Notes** and **MD&A** for number extraction. Trigger the KPI extractor only if a chunk (or its neighbors) trips **numeric cues** (currency/percent/units/EPS/YoY).
- For **auditability**, always store: `(value, unit, KPI_name guess, period anchor, section, docID, exact sentence span)` and return the **evidence sentenceID** with the answer.

3.4 Prompting patterns

- **Retrieval prompt**: seed with section label hints (e.g., “Prefer Item 7 (MD&A) when user asks about management’s analysis…”).

- **KPI prompt**: strict JSON schema with fields for `value`, `unit`, `period`, `as_of_date`, `evidence_sentenceID`; include a *refusal rule* if no explicit numeric evidence is present.
- **Narrative prompt**: cite `[sentenceID]` after each claim; instruct model to avoid deriving numbers—only restate or contextualize.

---

4) What to do with each CSV (practical use)

- **`section_label_suggestions.csv`** → load into a small mapping table for your pipeline; expose in the UI as human-friendly filters; use in query routing and eval slicing.
- **`top_ngrams_by_section.csv`** → build a simple **intent→section** lookup: when a query contains "risk", "adverse", "uncertainty", boost Risk Factors; for "revenue", "gross margin", boost MD&A/FS&Notes. This can be a dictionary + cosine over n-gram expansions.
- **`duplication_by_section.csv`** → set **per-section** dedupe thresholds (e.g., stricter in Items with boilerplate); also report **coverage after dedupe** to ensure recall isn't harmed.
- **`kpi_signal_scan_by_section.csv`** → configure **extractor budgets** (LLM calls/time) where signal is high; in low-signal sections, skip extractor and rely on narrative search only.

---

5) Section-wise expectations (policy you can codify)

- **Item 1 (Business/Overview)**: narrative heavy; useful for qualitative Q&A; numeric signal moderate (market/segment sizes appear occasionally).
- **Item 1A (Risk Factors)**: little structured KPI; high duplication potential across years (boilerplate); emphasize diversity + recency.
- **Item 7 (MD&A)**: rich numeric context (growth %, YoY, driver explanations); **top priority** for KPI+explanations pairing.
- **Item 8/Notes (Financial Statements & Notes)**: dense with currency/units; great for **audited KPIs**; tables often exceed normal chunk size → handle with table mode.
- **Controls/Exhibits**: low KPI value; keep for compliance/explanations but deprioritize for extraction.

(These align with your outlier check and the n-gram label suggestions.)

---

6) Evaluation slices you can build from here

- **By section label**: retrieval recall@k and answer correctness for MD&A vs Risk vs FS/Notes.
- **By period**: pre/post 2008, or rolling 5-year windows, to detect drift.
- **With/without dedupe**: show impact on recall and answer diversity.
- **KPI hit rate**: fraction of user KPI intents that produce a validated number+evidence (use your `kpi_signal_scan` to define eligible queries).

---

7) Immediate next steps (short list)

1. **Lock the section taxonomy**: freeze the mapping from `section` →label using your `section_label_suggestions.csv` (review a few sections manually).
2. **Implement section-aware retrieval**: add label boosts guided by `top_ngrams_by_section.csv`.
3. **Add dedupe at retrieval time**: diversity constraint over near-dupe pairs per `duplication_by_section.csv`.
4. **Gate the KPI extractor**: only run when numeric cues are detected (and in high-signal sections per `kpi_signal_scan_by_section.csv`).
5. **Table handling**: send table-like chunks to a separate path (either skip for narrative RAG or process with a table parser).

Citations to your notebook cells (provenance)

- N-gram signature creation & save (`top_ngrams_by_section.csv`)
- Duplication estimation & save (`duplication_by_section.csv`)
- KPI signal scan & save (`kpi_signal_scan_by_section.csv`)
- Section label suggestions & save (`section_label_suggestions.csv`)
- Dataset size and memory footprint (Polars printout)
- Temporal coverage table & stats
- Outlier (table-like) examples by section
- Section code distribution (0–19) and the need for mapping
- Company distribution/imbalance table & summary

REMEMBER THIS:

- ❌ Deep EDA on small_full before testing large_full
- Your section distributions WILL change
- Company balance WILL change
- Token stats might shift (if large_full has different companies/years)
- ❌ Assuming small_full is representative
- It's called "small" for a reason
- Likely a curated subset (e.g., only 10 companies, only 2002-2020)
- Large_full might include 100+ companies, 1990-2023, international filings
- ❌ Perfectionism on the wrong dataset
- Even if absolute numbers (counts, medians, token lengths) shift later, the qualitative shape of the data — structure, hierarchies, field types, sparsity, edge cases — rarely changes between the small and large splits.
  - The schema (cik, section, sentence, returns) is fixed.
  - The distribution form (e.g., some sections heavy, some sparse; certain companies dominating) will stay the same.
  - Anomalies like duplicates, boilerplate, table-like text are systemic, not random — they appear everywhere.