Term Project Checkpoint A

Greg Roberts and Flavio Mota

MSDS 459 - Knowledge Engineering

Apr 27, 2025

**1. What is your topic and why did you choose it?**Our project focuses on building a **consumer discretionary sector knowledge graph** that integrates rich relationships between publicly traded firms, their products, and observed consumer-spending patterns. We chose this for a variety of reasons including a strong interest in how consumer discretionary spending is closely linked to economic cycles. Greg works as a data scientist at a payments company where he has direct exposure to granular transaction data across small and medium-sized merchants, insights that are invaluable for understanding demand drivers in discretionary spending.

Embedding these data into a graph allows us to surface non-obvious links (for example, between macroeconomic indicators and sub-sector performance) and powers applications such as targeted stock recommendations or marketing-strategy optimization. This is done through “Named Entity Recognition,” (NER). Nodes, representing patterns gleaned from the web scraping process, are connected with lines (these lines are the edges).

These relationships are mined from the data using Relationship Extraction (“RE”).

Additionally, the knowledge graphs provide a visual representation of the available information and the relationships among the nodes. This visualization provides a useful framework that makes it incredibly easy to explain to an audience (e.g., the C Suite, stakeholders, investors, and so on).

**2. Initial thoughts on the database schema (EdgeDB graph-relational model)**Building on RDF Schema principles for ontology design, our EdgeDB schema will define object types (nodes) and links (edges) with both strong typing and graph semantics. Likely **node types** include:

* **Company** (ticker, headquarters, sector classification)
* **ProductCategory** (e.g., apparel, leisure equipment)
* **ConsumerSegment** (household income bands, demographics)
* **EconomicIndicator** (CPI, consumer confidence)
* **TransactionRecord** (timestamp, amount, merchant\_id)
* **DataSource** (BEA, BLS, FRED, FSBI, Wikipedia, Yahoo Finance, internal acquirer data)

The web scraping and data mining processes will likely reveal other nodes. And the new information will be used to adjust the algorithm to make the same robust and useful in gathering new data.

**3. Identified information sources and initial data collection** To populate the graph, we’ll combine:

* **Fiserv Small Business Index (FSBI)**: a rolling index that tracks small business revenue and consumer spending trends across key discretionary categories.
* **Government datasets**: consumer expenditure surveys (BEA), retail sales reports (U.S. Census Bureau), and labor statistics (BLS) to anchor our internal data to macro trends.
* **FRED series**: key consumer spending time-series (e.g., Personal Consumption Expenditures, Retail Sales) fetched directly from the Federal Reserve Economic Data API.
* **Web sources**:  
  + **Wikipedia** for structured lists (e.g., sector constituents, historical index values).
    - <https://en.wikipedia.org/wiki/Amazon>
    - <https://en.wikipedia.org/wiki/Home_Depot>
    - <https://en.wikipedia.org/wiki/AutoZone>
    - https://en.wikipedia.org/wiki/Booking\_Holdings
    - https://en.wikipedia.org/wiki/McDonald%27s
    - https://en.wikipedia.org/wiki/Nike,\_Inc.
    - https://en.wikipedia.org/wiki/TJX\_Companies
    - https://en.wikipedia.org/wiki/Lowe%27s
    - https://en.wikipedia.org/wiki/Mercado\_Libre
  + **Yahoo Finance** APIs for real-time price and fundamental data on consumer discretionary stocks.
* **Company disclosures**: SEC EDGAR filings (10-Ks, earnings calls) for firm-level fundamentals and strategic initiatives.
* **Industry publications and news feeds**: trade journals and targeted web crawls (using a focused-crawler pipeline with distiller modules) to capture qualitative context.

**Automated ingestion:**We’ll implement **Scrapy** spiders to scrape HTML tables and JSON endpoints from FSBI, FRED, Wikipedia, and Yahoo Finance; ingest SEC filings with BeautifulSoup/XML parsers; and pull CSV/Excel data via Python ETL scripts. Extracted data—whether JSON, CSV, or HTML—is then normalized and upserted into EdgeDB. For unstructured documents (PDFs and web pages), we’ll integrate LangChain loaders and text-splitters to extract and map facts into nodes and edges.

**4. Likely users and user questions** Primary users include:

* **Equity analysts and portfolio managers**, querying “Which consumer discretionary firms show leading growth among mid-income segments?”
* **Marketing strategists**, asking “Which product categories correlate with upticks in disposable-income spending during holidays?”
* **Data science teams**, performing advanced analytics such as “What sub-sector clusters emerge when segmenting firms by both transaction velocity and average ticket size?”

By traversing the knowledge graph, users can quickly answer multi-hop questions (e.g., identify companies whose stock prices align with shifts in consumer confidence), rather than assembling data manually.

**5. Proposed application and KG utility** We envision a **web-based dashboard and API** that layers:

1. **Graph-powered search & exploration**: interactive graph visualizations for exploratory analysis.
2. **Natural-language question answering**: a RAG-augmented chatbot that translates user queries into graph queries for on-the-fly insights.
3. **Recommendation engine**: using graph analytics (e.g., node centrality, community detection) to surface high-potential stock picks or marketing targets.

In this stack, the knowledge base underpins information retrieval (via schema-driven queries), information extraction (through automated ETL and Scrapy-based ingestion), question-answering (via embedding-augmented retrieval over node properties), and recommendations (through graph algorithms that detect patterns in spending behavior and firm fundamentals).