Knowledge Graphs for financial text extraction

Hannah Tollié
Prof. Ali Hirsa; Advisor: Miao Wang

Introduction

Financial markets are continuously influenced by events like earnings reports, regulatory changes, and macroeconomic announcements. Much of this information is embedded in unstructured text, making systematic analysis a challenge.

Traditional NLP techniques, such as sentiment analysis, often fail to capture the nuanced relationships between financial entities and events. To overcome this, our project leverages large language models (LLMs) to extract structured SPO triplets—**(entity1, relation, entity2)**—from financial headlines to form the basis of knowledge graphs (KGs). These graphs track how relationships change over time and serve as inputs for predictive modeling via graph neural networks (GNNs).

Expanding on different LLM, and testing several pipeline organization to optimize triplet extraction., We then standardize entities and relations, and slicing the graph into enty types. This framework enables more comprehensible data visualization.

Phase 1: Model Test for Triplets Generation

We began by extracted SPO triplets with LLama3.1, by for bypassed by other model since, the project started. We tried many but observed that Llama4-Scout and Gemini2.5-Flash were extracted the best triplets. Furthermore, Llama4 is the most efficient model for the same number of input tokens.

Model Name	Input Tokens	Extraction Time (s)
meta-llama/llama-4-Scout-17b	1008	6.89122057
openai/gpt-oss-120b	1008	143.9340577
gemini-2.5-flash	1008	163.8646295
qwen/qwen3-32b	1008	329.6997974

Phase 2: Prompt Test for Triplets Generation

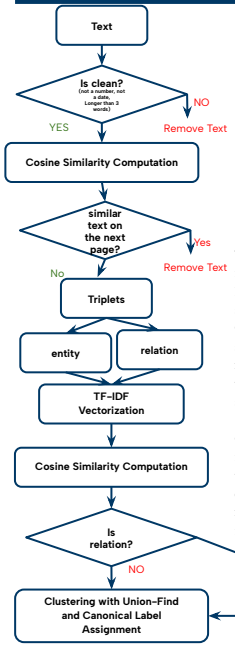
Prompt engineering becomes increasingly important and can make an LLM response vary greatly. To take advantage of the situation, there was a need to test the SPO-triplet extraction accuracy for each prompt organization. How can we arrange the different parts of a prompt to extract the best answers.
All the extractions of this part were performed with Llama3.1-8B.

Prompt Configuration	Precision	Recall	F1-Score	Avg. Similarity
V_0_Zero_Shot	1.00000	0.14286	0.14286	0.14286
V_1_SUM	0.30952	0.19048	0.17460	0.52092
V_2_SSP	0.40476	0.19048	0.17460	0.42650
V_3_ESP	0.45238	0.19048	0.17460	0.44194
V_4_EUM	0.34921	0.19048	0.16667	0.53157

Here, the V0 prompt contains no example; V2, and V3 prompts place the examples after the user prompt respectively before, and after the main tasks and instructions; V1 and V4 prompts place the example respectively at the beginning and at the very end of the user message.

The results suggest that, **adding an example has a strong impact** on the LLM response accuracy. Additionally, putting the prompt around the user prompt seems to generate more precise SPO – more similar SPO by words extracted –. However, putting the examples toward the beginning or the end of the user message seems to also increase response's (although slightly less) and increase precision and significantly increase the Average Similarity score – extract more semantically similar SPO –. Hence the V4 prompt seems to have the **best Precision Average_Similarity_Score combination.**

Phase 3: Data Cleaning Validation & Refinement



(1) Entity-level filtering and outlier removal

Temporal Pattern Filtering - we used regular expressions to detect and exclude: Years (like "1999" or "2022") / Fiscal quarters (e.g., "Q1", "Q4") / Named months and full date formats (e.g., "January", "2024-03-01", "03/22")

Low-Frequency Entity Pruning - We removed entities that appeared very rarely—fewer than five times in the entire dataset. Entities that only show up once or twice are often typos, irrelevant phrases, or just noise.

Rule-Based Validity Checks - We applied a few simple rules to catch and remove clearly invalid entities. We excluded any entity that: Was just a number (like "12.5") / Was just a symbol (like "%") / Was a short fiscal abbreviation (like "FY" or "Q3")

(2) Neural Networks Definitions

To predict links within the knowledge graph, we evaluate two primary inductive Graph Neural Network (GNN) architectures. To ensure that learning is based purely on the graph's structure, both models use only the scalar node degree as an input feature.

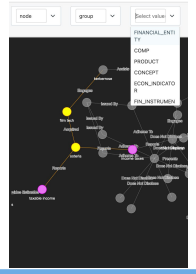
Graph Attention Network (GAT): The GAT architecture incorporates a self-attention mechanism, allowing it to weigh the importance of neighboring nodes when creating embeddings. Our implementation uses a two-layer network: the first layer has four attention heads to enhance stability, and the second layer uses a single head to produce the final 16-dimensional node embeddings.

GraphSAGE: This model generates node embeddings by aggregating features from a node's local neighborhood. We use the mean aggregator variant, where embeddings are updated by taking the mean of neighbor embeddings from the previous layer. Our architecture is also a two-layer network with a 16-dimensional final embedding, which is highly effective for generalizing to unseen nodes.

Clustering - we applied a **union-find algorithm**, which forms clusters by transitivity. If "A" ~ "B", and "B" ~ "C", then "A", "B", and "C" will be placed in the same cluster, even if "A" and "C" weren't directly matched.

Phase 4: Dynamic Visualization

Zoom-in view of nodes from COMP



The knowledge graph visualization enables quick analysis of financial events and relationships by transforming data extracted by Llama3 into KGTK-compatible formats. Each extracted triplet is standardized into a relational schema (Entity1 - Relation - Entity2). Our updated pipeline generalizes this process by enabling segmentation on **entity type.** For each user-defined time range, the dataset is filtered by date, and a corresponding edge–node pair is generated. To render the graph interactively and enable temporal analysis, we used **NetworkX** to construct a directed graph based on the generated edge-node data and **Pyvis** to visualize it in-browser.

Phase 5: Evaluation

We evaluated the Graph Neural Network (GNN) architectures— **Graph Attention Network** and **GraphSAGE**—using both extracted triplets from previous method with previous prompt and the new method. Our qualitative assessment focused on:

- Oldest method demonstrated higher metrics but a great instability. Overall this model seems to suffer from overfitting and poor reliability-

- Newest Method: (only train SPO triplets extracted from 32 texts) is definitely more promising, while metrics demonstrate a lower final fit, we get a more stable training phase and more realistic outputs.

Quantitative comparisons are summarized visually in the adjacent performance metrics table.

Metric	Previous Extraction	Soteria Extraction
Last MRR	0.3642	0.2617
Last H@1	0.1598	0.0385
Last H@10	0.9907	0.8462
Last MAE	146 835.2	858.9
Last Loss	0.0017	0.0046
Training Stability	*Low*	*High*

Results

This project demonstrates that the quality of Subject-Predicate-Object (SPO) extraction greatly influences the quality of downstream graph-based predictions. Our findings indicate that among the models tested, Gemini 2.5 Flash and Llama 4 are currently the most effective for the SPO extraction task.

Additionally, this study examines how prompt organization affects output quality. The results show that

1. SYSTEM_INSTRUCTIONS
2. RULES
3. ENTITY_TYPES
4. OUTPUT_FORMAT
5. input_text
6. META_INSTRUCTION
7. FEW_SHOT_EXAMPLES

appears to be the most efficient design for an SPO extraction prompt.

Our interactive Knowledge Graph (KG) visualizations provided dynamic, intuitive insights into financial relationships displaying influential entities and their interdependencies, and visualizing influence, novelty, and entity impact to offer a comprehensive understanding of market-moving factors. High-influence entities corresponded closely to significant real-world market events and company announcements. Creating an clear and comprehensible graphical representation supporting the models trained in the project.

Reference

- Minhao Pang Kaidi Chen, Haining Han. (2025). LLM and Graphs for Financial Texts. IEOR EIB 4, Columbia University.
- Kwesi Cobbina and Tianyi Zhou. Where to show demos in your prompt: A positional bias of in-context learning. arXiv, Jul 2025.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks. International Conference on Learning Representations (ICLR).
- L Brandon Wang Sukjun Hwang and Albert Gu. Dynamic chunking for end-to-endhierarchical sequence modeling. ArXiv, Feb 2025.