DYNAMIC TAXONOMY CONSTRUCTION AND THEMATIC FILTERING FOR FINANCIAL KNOWLEDGE GRAPHS WITH LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Financial analysis and investment strategies have long played a central role in both academic research and industry practice. The rise of advanced algorithms and Artificial Intelligence (AI) has made understanding financial markets and making informed investment decisions increasingly complex, particularly due to the integration of multifaceted data sources such as text, numerical data, and domain-specific knowledge. Taxonomy provide essential frameworks for organizing financial concepts, while knowledge graphs (KGs) interlink diverse entities such as companies, market events, and economic indicators enabling sophisticated analytics and reasoning.

This paper presents a framework that leverages Large Language Models (LLMs) to dynamically construct taxonomy and financial knowledge graphs from unstructured sources, specifically daily conference call transcripts. Our approach addresses the high variability in how financial entities and relationships are described by mapping extracted node types and relations to their parent nodes within adaptive taxonomy. This process reduces semantic variation and produces more consistent, manageable knowledge graphs.

A key innovation of our method is the application of theme-based and thematic filtering, which organizes information by relevant topics and enhances the efficiency of data analysis and search for investors. By classifying and filtering graph data according to themes, our framework significantly reduces complexity and variability, making financial information more accessible and actionable. Empirical results demonstrate that this approach improves graph search efficiency by 6% and enables the extraction of richer insights from financial data, supporting more effective decision-making in dynamic market environments.

1 INTRODUCTION

Financial analysis and investment strategies have long been pivotal in both academic research and industrial applications. With the advent of advanced algorithms and Artificial Intelligence (AI), understanding financial markets and making informed investment decisions has become increasingly complex. Historically, key challenges have included accurate prediction of market trends, efficient data processing, and integration of multifaceted data sources such as text, numerical data, and domain-specific knowledge. Taxonomies serve as foundational frameworks for describing and structuring financial concepts, enabling hierarchical organization and clustering of information.

Knowledge Graphs (KGs) provide a powerful mechanism for interlinking diverse pieces of financial information, facilitating sophisticated analytics and reasoning. In the financial domain, these graphs can model complex relationships between entities such as companies, stocks, market events, and economic indicators. Leveraging taxonomies helps prevent the proliferation of semantically similar relationships between entities, thus maintaining the quality and consistency of data within the graph. However, the variability in terminology and phrasing, especially in unstructured sources such as daily conference call transcripts, complicates the process of filtering and identifying relevant information.

Our research introduces a novel framework that applies Large Language Models (LLMs) to dynamically construct taxonomies and financial knowledge graphs from unstructured financial data. By mapping extracted node types and relations to their parent nodes within adaptive taxonomy, we reduce semantic variation and improve graph consistency. A key innovation of our approach is the integration of theme-based and thematic filtering, which organizes information by relevant topics and enables more effective filtering and retrieval of financial insights. Thematic filtering allows users to focus on specific areas of interest, improving both the consistency and accessibility of the knowledge graph.

In summary, our work builds on foundational concepts in financial data organization and knowledge graph construction, while introducing a dynamic, theme-driven approach to address the challenges of high variability in financial data extraction and filtering. This methodology aims to enhance the efficiency and analytical value of financial data analysis, supporting more informed decision-making in complex and rapidly evolving market environments.

2 Related work

Pérez et al. (Perez et al., 2006) provide a foundational formal analysis of SPARQL, the standard query language for RDF. They define the compositional semantics of SPARQL graph patterns, establish normal forms, and rigorously analyze the computational complexity of query evaluation, demonstrating that SPARQL query evaluation is PSPACE-complete in general. Their work clarifies the expressiveness and limitations of SPARQL when operating on static, predefined ontologies. Building on this foundation, Arenas et al. (Marcelo Arenas, 2016) develop a theoretical framework for faceted search of RDF-based knowledge graphs, focusing on systems enhanced with OWL 2. They identify SPARQL fragments suitable for faceted interfaces, analyze query complexity, and propose efficient algorithms for interface generation and update, all within the constraints of ontology-guided navigation.

Subsequent research has explored ways to make ontologies and knowledge graphs more adaptive and data-driven. Zhou et al. (Zhou et al., 2022) introduced methods to reshape domain ontologies into compact, data-oriented structures to construct KGs from relational tables. Tatchukova and Qu (Tatchukova & Qu, 2024) proposed bounding edge embeddings to regulate KG growth under host ontologies. Kommineni et al. (Kommineni et al., 2024) automated ontology and KG construction from competency questions with minimal expert involvement. Yu et al. (Yu et al., 2024) combined financial KGs with LLMs for causal reasoning, while Cauter and Yakovets (van Cauter & Yakovets, 2024) and Shimizu and Hitzler (Shimizu & Hitzler, 2024) investigated LLMs for ontology-guided extraction and modular modeling. Abolhasani and Pan (Abolhasani & Pan, 2024) presented OntoKGen, an LLM-driven pipeline for ontology extraction and KG generation, and Ding et al. (Ding et al., 2024) proposed a type-aware model that emphasizes relation updates derived from Wikipedia-based ontologies.

Across this literature, the dominant paradigm remains ontology-driven: expert-curated hierarchies such as FIBO, Schema.org, YAGO, or Wiki Finance provide type-aware indexing, query expansion, and logical entailment. Although effective, these static ontologies face challenges in domains like finance, where terminology evolves rapidly and semantic variability is high.

Our work shifts the focus to dynamic taxonomy-driven organization. Rather than relying on fixed, expert-defined one, we dynamically generate adaptive taxonomy from unstructured financial data using large language models (LLMs). This taxonomy reduce semantic variation by mapping diverse expressions to parent concepts, producing more consistent and precise knowledge graphs. Unlike query expansion approaches that prioritize recall, our taxonomy-driven method emphasizes precision and interpretability in noisy, fast-changing domains. This adaptive, data-driven framework provides a practical alternative to taxonomy-centric methods, aligning knowledge graph construction more closely with real-world financial discourse and market dynamics.

Table 1 summarizes the key differences between our approach and existing methods.

Approach	Ontology/Taxonomy Type	Query Method	Variability Handling
RDF with RDFS/OWL	Static, predefined	SPARQL with entailment	Subclass inference
Hierarchical Faceted	Fixed taxonomies	Faceted browsing	Hierarchical navigation
YAGO/ DBpedia	Large static KB	SPARQL + type hierarchy	Type-based filtering
Our Approach	Dynamic, data-driven	Vector similarity + themes	Compression via parent mapping

Table 1: Comparison of our approach with existing methods

3 METHODOLOGY

3.1 TAXONOMY DEFINITION AND MAPPING

The taxonomy T is a structured mapping of specific node types and relations to their parent nodes. It is defined as:

$$\mathcal{T} = (x, f(x)) \mid x \in X \tag{1}$$

Here, X is the set of all possible node types and relations, and $f(x)$ is the parent node for x. The mapping function $f : X \to P$ assigns each node type or relation x to its parent node $f(x)$, where P is the set of parent nodes.

The ideal host relation and node type taxonomies may be generated from an oracle context such as Wiki Finance (Ding et al., 2024). In the absence of a domain-specific knowledge base, we can utilize our initial knowledge graph relations and node types, leveraging a large language model (LLM) to review nodes and propose parent node types and relations. In this approach, the host taxonomy is structured as a hierarchical system with statistical mappings, rather than as a formal standard template like OWL. This lightweight method enables rapid adaptation and low maintenance, which is essential for fast-evolving financial domains.

The goal is to facilitate data integration by establishing host taxonomies that define parent nodes relevant to the financial sector. For example, the taxonomy might include parent nodes such as Executive for roles like “CEO” and “CFO,” and Financial Event for events like “Earnings Call” and “Dividend Announcement.” The host taxonomy is subject to continuous monitoring and updates, incorporating expert domain knowledge.

They are indexed using a Retriever system, where each row from the taxonomy depth search contributes to the Retriever index. Subsequent knowledge graph (KG) relations and node types are submitted as queries to the Retriever taxonomy, facilitating the extraction of the most relevant match. A similarity threshold of 60% is applied, ensuring that node types and relations exceeding this threshold are linked to their respective parent nodes within the taxonomy. This process standardizes similar node types and relations, thereby reducing variability. Queries that do not yield results are collected and transformed into a new taxonomy. The new taxonomy is monitored by knowledge experts before merging with the host taxonomy (see Appendix D.2).

Later, the granular relation is embedded along with the node entity, while the higher-level relations are stored in metadata to enable easy data filtering and searching. Figure 1 shows two samples of relation and node type taxonomies. For more details, please see Appendix B.

3.2 GRAPH CONSTRUCTION AND TRANSFORMATION (THEMATIC GRAPH)

In the construction of a financial knowledge graph, the process begins with the extraction of statements that have a significant financial impact. Each statement, which may consist of one or more sentences, is analyzed to generate multiple triplets.

A triplet is a fundamental unit that represents two entities (i.e., nodes), known as the ”head” and ”tail,” connected by a ”relation”. Additionally, each entity is characterized by its type, providing semantic context and meaning. For instance, in a financial knowledge graph, a triplet might consist

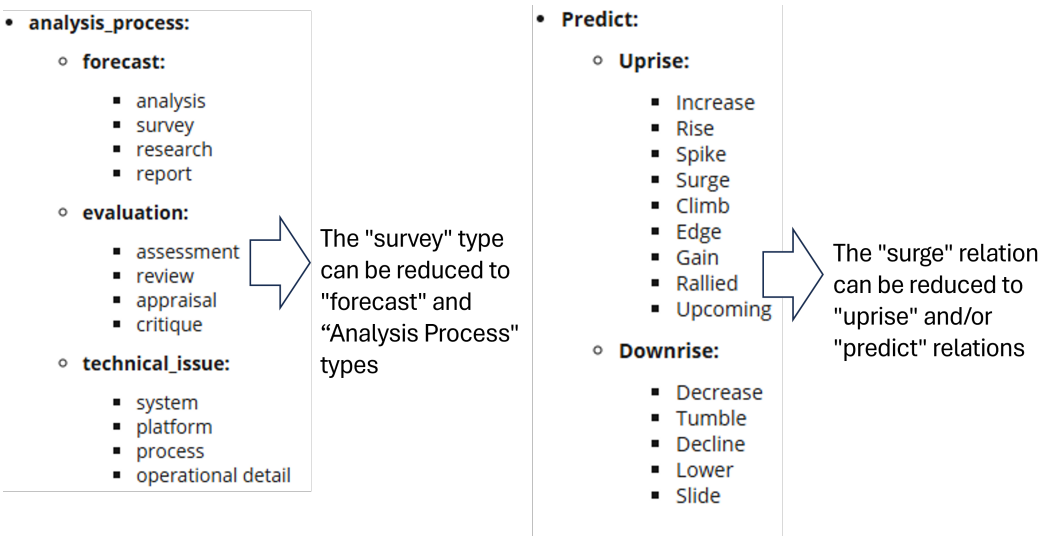

Figure 1: Example taxonomies for node types (left) and relations (right).

of the head entity ”Company A,” the relation ”acquired,” and the tail entity ”Company B,” with the node types being ”Corporation.” The graph generation process involves systematically extracting these triplets from the impactful statements and organizing them into a structured format.

We construct the initial dynamic graph, G_D°, from the document using extracted entities, their types, and relations:

$$G_D^\circ = \{(e_i, r_{ij}, e_j) \mid e_i, e_j \in E^\circ, r_{ij} \in R^\circ\} \tag{2}$$

Here, e_i and e_j are entity types (i.e., head and tail), and r_{ij} is the relation between them. E° is the set of all entity types extracted from the chats, while the dot notation indicates that it is the initial or base set of the initial dynamic graph. R° denotes the set of all relations extracted from the chats. This process is further refined by incorporating taxonomies, which map specific node types and relations to their parent nodes, thereby normalizing the graph structure by adding a theme to each triplet. By capturing intricate relationships, ensuring consistency, and effectively translating complex textual information into actionable insights, the knowledge graph becomes a powerful tool for data analysis and decision-making in the financial sector.

Given (e_i, r_{ij}, e_j) triplets of graph G_D°, we apply taxonomies by replacing nodes with their parent nodes from the taxonomy:

$$(e_i', r_{ij}', e_j') = (f(e_i), f(r_{ij}), f(e_j)) \tag{3}$$

The transformed graph G_D is composed of triplets with a theme for each node type and relation:

$$G_D = \{(e_i', r_{ij}', e_j') \mid (e_i, r_{ij}, e_j) \in G_D^\circ\} \tag{4}$$

where $V(G_D) < V(G_D^\circ)$. Here, V(S) measures the variability of a graph S. This could be based on the diversity or entropy of node types and relations (see appendix B) . Table 2 shows an example of theme triplet.

	Head	Head type	Tail	Tail type	Relation
Triplet	US Administration	Organization	Tariff	Economic Policy	Offset
Theme	-	Organization	-	Policy	Policy

Table 2: Example of theme triplet

4 GRAPH SEARCH

4.1 THEMATIC RETRIEVER

Retriever is the most important part of the Retriever pipeline. The higher-level relations and types are stored as metadata representing the theme of each triplet, allowing us to filter them based on the topic of interest. During retrieval, we filter the data using the candidate themes T_q to increase the accuracy of retrieval. Let D be the set of all data entries, and D_T be the subset of data entries relevant to themes T_q:

$$D_T = \{d \in D \mid \text{themes of } d \cap T_q \neq \emptyset\} \tag{5}$$

A retriever is used to extract the most relevant data entries from D_T based on the question q. A retrieval function $R(q, D_T)$ is defined to return the top matching data entries:

$$R(q, D_T) = \{d^* \mid d^* \text{ is the top match for } q \text{ in } D_T\} \tag{6}$$

The filtering process aims to enhance retrieval accuracy by ensuring that only data entries relevant to the classified themes are considered, thereby reducing noise and improving precision.

4.2 THEMATIC PATH SEARCH

In the context of a query, the path search seeks to identify connections between two entities, such as swap cases between two currencies (i.e., nodes). The process begins with an initial node $e_1 \in E$ and concludes with a second node $e_2 \in E$ that is most relevant to the query. For each node e and edge r encountered during traversal, we apply taxonomy mapping $e' = f(e), \quad r' = f(r)$. This ensures that the traversal path reflects the thematic concept for each triplet. During traversal, nodes and edges are filtered based on theme classification. If $T(e') \cap T_q = \emptyset$, skip e', where T_q is the set of themes relevant to the query. The traversal path P_t is the sequence of nodes and edges that meet the taxonomy, theme:

$$P_t = \{(e', r') \mid e' \in E, r' \in R, f(e), f(r), T(e') \cap T_q \neq \emptyset\} \tag{7}$$

This structured approach ensures that the path search is aligned with the thematic and taxonomy framework, enhancing the relevance and precision of the results in the context of financial knowledge graphs.

5 EXPERIMENT

In our experiments, we construct a pipeline to extract statements with financial impact, convert them into triplet sets for relationship mapping, and refine these relationships through taxonomy layers, reducing unique values and enhancing thematic consistency. We evaluate our retriever with LLM-generated queries, showing improved precision and F1-score using the Theme based GraphRAG model.

5.1 KNOWLEDGE GRAPH CONSTRUCTION

We process around 1000 financial messages daily, transforming them into statements highlighting financial impacts in macroeconomic and credit assets like foreign exchange options, swaps, bonds, and commodities. Using GPT-4o and LoRA finetuned Mistral-7B-Instruct-v0.2 models, we extract self-contained statements reflecting market outlooks, excluding non-relevant content. For details on prompts and hyperparameters, see Appendix C. The performance of the fine-tuned model is evaluated against results obtained using GPT-4o on a subset of 40 documents (Table 3). An annotation team (appendix F) evaluated statements on four metrics: dependency, relevancy, hallucination, and missing statements. GPT-4o excelled in all metrics.

		GPT-4o (%)	Mistral (%)
Dependency	Self-contained	86	70
Relevancy	Relevant	95	83
Hallucination	Strong	0	1
	Mild	2	0
Missing statements		16	29

Table 3: GPT-4o vs. Mistral model in statement extraction

		Accurate(%)	Semi-Accurate(%)	Inaccurate(%)
GPT-4o	Head	93	5	2
	Tail	83	13	4
	Relation	82	13	5
	Average	**86**	**10**	**4**
Mistral	Head	72	23	5
	Tail	44	31	25
	Relation	56	26	18
	Average	**57**	**27**	**16**
Flan T5	Head	75	18	7
	Tail	42	16	43
	Relation	39	22	39
	Average	**52**	**18**	**30**

Table 4: GPT-4o vs. Mistral and Flan T5 in triplet extraction (96 samples)

In phase two, we convert extracted statements into triplet sets of (head , head_type), (tail , tail_type) and relation. We use the Flan T5 large model for LoRA fine-tuning and consider the Mistral model for integration. Performance is evaluated against GPT-4o (Table 4), with details in Appendix C. Triplet extraction is manually annotated as accurate, semi-accurate, or inaccurate, based on entity alignment and detail inclusion. Here's an evaluation example:

- **Text:** Egypt's cash traded well recently with IMF fund approval, despite positioning issues.
- **Triplets:** (Head: Egypt Cash, Tail: IMF Approval, Relation: Traded well recently)
- **Evaluation:** (Head: Accurate, Tail: Accurate, Relation: Semi-accurate; suggest: Trade well)

The evaluation results show GPT-4o can extract the triplets with more than 80% accuracy while Flan T5 shows lowest performance with failing to extract the triplets around 30%.

5.2 Taxonomy Construction

In this experiment, we evaluated the integration of two taxonomy layers—parent and grandparent—for both entity types and relations within our knowledge graph. The evaluation was conducted on a dataset of 1,780 triplets using GPT-4o (see Appendix D for details).

Relation mapping is context-dependent; for example, the relation decrease maps to predict when referring to future events, but to performance when describing past events. To ensure precise mapping, we incorporated relation type during triplet generation, allowing context-aware alignment with taxonomy nodes (see Appendix C, Figure 6).

As shown in Table 5, the initial 944 unique relation values were reduced to 49 at the parent layer, and further collapsed to 8 categories at the grandparent layer. This hierarchical reduction illustrates how taxonomies effectively normalize variability while preserving semantic distinctions.

Table 6 presents the Grandparent categories for Relation and Type, designed as themes within metadata to aid investor filtering.

We evaluated the taxonomy mapping by providing the annotation team with 300 samples of relation and node types, focusing on categorizing accuracy as correct or inaccurate. Table 7 shows the accuracy ratio for each layer, with over 80% accuracy for grandparent layer in both relation and

Node Property	Node Frequency		
	Grandparent layer	Parent layer	Child layer
Node Type	18	60	2403
Relation	8	49	944

Table 5: Type and relation frequency across taxonomy layers

	Themes (Grandparent layer)
Relation	Performance, Predict, Policy, Market, Assessment, Causal Factor, Strategy, Communication
Type	Resource, Person, Region, Analysis, Strategy, Event, Temporal, Communication, Legal, Process, Investment, Assessment, Economic, Politic, Market, Organization, Finance, Policy

Table 6: Grandparent taxonomy for types and relations

node type mapping. The inaccuracies for grandparent layer were due to strong topic overlap in certain statements.

	Grandparent %	Parent %	Child %
Relation	82	60	68
Node Type	84	68	-

Table 7: Evaluation of taxonomy mapping accuracy

Figure 2 illustrates a theme based subgraph with 22 relations, featuring only 6 unique themes: Predict, Strategy, Policy, Market Movement, Assessment, and Causal Factor.

5.3 Theme Retriever

We evaluated our retriever by generating 80 questions with a Large Language Model (LLM), then paraphrasing and testing them against queries. Each query was mapped to 1-3 relevant categories from our theme list or labeled as NA, simulating finance experts'decision-making. Queries were processed through our vector index with and without theme-based filtering. Table 8 shows question samples, while Table 9 details results, highlighting the Theme based GraphRAG model's superior precision and F1-score for the @5 test across most themes. The assessment used 1,493 unique statements with the sentence-transformers/all-MiniLM-L6-v2 model (Reimers & Gurevych, 2019) for our retriever.

To assess the statistical significance of our results, we combined all theme results, as each theme included only 10 questions. Table 10 summarizes the significance tests:

Our thematic approach is particularly effective for domain experts seeking precise insights from complex, highly variable, and rapidly evolving financial data. It excels in scenarios that require thematic filtering, enabling focused and context-specific retrieval (see Table 8). For general or broad queries, however, standard RAG systems may outperform our method, as thematic filtering offers less advantage in such cases. Additionally, the effectiveness of thematic search depends on optimal theme selection, which can be limited when relying solely on LLMs rather than domain experts. Importantly, our system is designed to complement rather than replace existing methods, allowing users to choose the most appropriate tool for their specific needs. Combining both systems to leverage their respective strengths is a promising direction for future research.

To ensure reproducibility, we conducted the test using publicly available data. Detailed information about the test is available in Appendix E.

5.4 Path Search

In a second search experiment, we conducted a path search for a swap query to further assess our framework's capabilities. This method involves identifying head and tail entities within the query

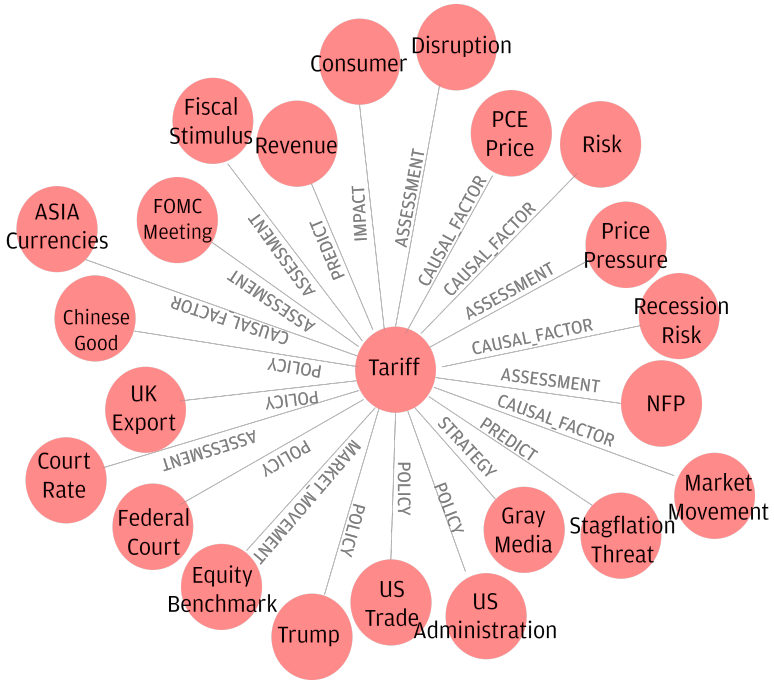

Figure 2: Theme subgraph for Tariff Node

Themes	Question
PERFORMANCE	What was the performance of Wayfair in a challenged category?
PREDICT, CAUSAL FACTOR	What is the expected impact of import front-loading on 2Q growth forecasts?
EVENT, ECONOMIC, MARKET	How are recent geopolitical events and economic data affecting the US dollar and market sentiment?
PERFORMANCE, TREND, MARKET	How does the performance of EM corporate credit compare to US credit given current spread levels?

Table 8: Examples of questions labeled by Theme

	RAG			GraphRAG		
Topic	**Precision**	**Recall**	**F-score**	**Precison**	**Recall**	**F-score**
Performance	0.5	**0.75**	**0.6**	**0.51**	**0.75**	**0.60**
Predict	0.31	0.87	0.45	**0.52**	**0.92**	**0.67**
Policy	0.54	**0.98**	**0.69**	**0.58**	0.86	**0.69**
Market Movement	0.46	1.00	0.63	**0.53**	1.00	**0.70**
Assessment	0.68	**0.97**	**0.80**	**0.73**	0.74	0.73
Causal Factor	0.70	**0.98**	**0.82**	**0.71**	0.81	0.76
Strategy	0.51	0.75	0.61	**0.64**	0.80	**0.71**
Communication	0.31	0.85	0.46	**0.36**	**0.90**	**0.51**
Average	0.50	**0.89**	0.63	**0.57**	0.85	**0.67**

Table 9: Performance comparison between RAG (without Theme filterig) and GraphRAG (with Theme filtering) retrievers

Metric	t-statistic	p-value	Significant at 90%?	Significant at 95%?
Precision	3.323	0.001	Yes	Yes
Recall	-1.719	0.090	Yes	No
F-score	1.798	0.076	Yes	No

Table 10: statistical significance test for Table 9

Path	Rank Weight	Quote
$PLN \rightarrow chargelower \rightarrow HUF$	100.0	**PLN** and **HUF** lead the **charge lower**
$EURPLN \rightarrow Volatility \rightarrow FXOs \rightarrow Vols \rightarrow HUF$	33.3	- **EURPLN** fluctuations initially drive **volatility** to increase sharply, reflecting on market instability. - **FXOs** adjustments stabilize overall **Vols** to decrease and steepen the curve as strategies shift. - **HUF** movements sustain elevated **Vols** level, maintaining market uncertainty above 400.

Table 11: Example of path, rank weight, and quote

and searching for fuzzy matches among potential candidates. We extract all feasible paths connecting these candidates and rank them using the formula:

$$\text{Rank Weight} = \frac{\text{HS} \times \text{TS}}{\text{Path\_length}} \tag{8}$$

where HS is the similarity of the head node to the query start node, TS is the similarity of the tail node to the query end node, and Path_length is the number of edges in the path. Paths and edges are ranked, with additional filters applied based on theme and/or rank weight. The final selection is summarized into quotes for a concise overview of findings. In this test, no theme was set as theme search is still under investigation. Below is an example of our search:

Trade Summary:"Pay fixed rate on PLN 5y and Receive fixed rate on HUF 5y"

Two possible paths within the selected time span are shown in Table 11

6 Conclusion

The integration of knowledge graphs, powered by large language models (LLMs), offers a robust framework for managing and analyzing unstructured data within the financial sector. Our approach effectively addresses the challenge of variability in extracted triplets comprising head nodes, tail nodes, and relations and node types by employing a taxonomy to standardize these elements. This standardization results in a more consistent and applicable knowledge graph, enhancing both its searchability and the quality of insights derived from the data.

7 Limitations

The semi-automated taxonomy construction process requires expert input for validation and refinement. Relying solely on LLMs for taxonomy generation can lead to uncontrolled growth. To mitigate this, we advocate for a human-in-the-loop approach during each merge into the host taxonomy. This process allows domain experts to prune unnecessary branches, effectively controlling the taxonomy's size and ensuring its relevance.

Additionally, we propose tagging the taxonomy with dates to discard nodes that have not been referenced for over certain time (6 months in our use-case). This strategy ensures the taxonomy remains current by removing outdated data, thereby preventing excessive growth.

The taxonomy builder exhibits a strong bias towards node suggestions provided in the prompt. It is essential to revise these suggestions for each new use case to ensure accuracy and relevance.

REFERENCES

Mohammad Sadeq Abolhasani and Rong Pan. Leveraging llm for automated ontology extraction and knowledge graph generation, 2024. URL https://arxiv.org/abs/2412.00608.

Amazon Web Services. Amazon ec2 g5 instances. https://instances.vantage.sh/aws/ec2/g5.xlarge, 2025. Accessed: 2025-05-18.

Linyi Ding, Sizhe Zhou, Jinfeng Xiao, and Jiawei Han. Automated construction of theme-specific knowledge graphs, 2024. URL https://arxiv.org/abs/2404.19146.

Vamsi Krishna Kommineni, Birgitta König-Ries, and Sheeba Samuel. From human experts to machines: An llm supported approach to ontology and knowledge graph construction, 2024. URL https://arxiv.org/abs/2403.08345.

Wenhao Liang, Zhengyang Li, and Weitong Chen. Enhancing financial market predictions: Causality-driven feature selection, 2024. URL https://arxiv.org/abs/2408.01005.

Evgeny Kharlamov Šarūnas Marciuška Dmitriy Zheleznyakov Marcelo Arenas, Bernardo Cuenca Grau. Faceted search over rdf-based knowledge graphs. *Web Semantics*, pp. 55–74, 2016.

Jorge Perez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql, 2006. URL https://arxiv.org/abs/cs/0605124.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

Cogan Shimizu and Pascal Hitzler. Accelerating knowledge graph and ontology engineering with large language models, 2024. URL https://arxiv.org/abs/2411.09601.

Kina Tatchukova and Yanzhen Qu. Restricting the spurious growth of knowledge graphs by using ontology graphs. *IEEE Access*, 12:12797–12807, 2024. doi: 10.1109/ACCESS.2024.3355457.

Zeno van Cauter and Nikolay Yakovets. Ontology-guided knowledge graph construction from maintenance short texts. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pp. 75–84, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.kallm-1.8. URL https://aclanthology.org/2024.kallm-1.8/.

Ziqi Yan, Jiqiang Liu, Gang Li, Zhen Han, and Shuo Qiu. Privmin: Differentially private minhash for jaccard similarity computation, 2017. URL https://arxiv.org/abs/1705.07258.

Guanyuan Yu, Xv Wang, Qing Li, and Yu Zhao. Fusing llms and kgs for formal causal reasoning behind financial risk contagion, 2024. URL https://arxiv.org/abs/2407.17190.

Dongzhuoran Zhou, Baifan Zhou, Zhuoxun Zheng, Egor V. Kostylev, Gong Cheng, Ernesto Jimenez-Ruiz, Ahmet Soylu, and Evgeny Kharlamov. Enhancing knowledge graph generation with ontology reshaping. *Lecture Notes in Computer Science (LNCS)*, 13384:299–302, 2022. URL https://doi.org/10.1007/978-3-031-11609-4_45.

A APPENDIX

B THEMATIC GRAPH PIPELINE

This appendix provides an in-depth overview of the theme based search pipeline and the process of theme generation through taxonomy mapping.

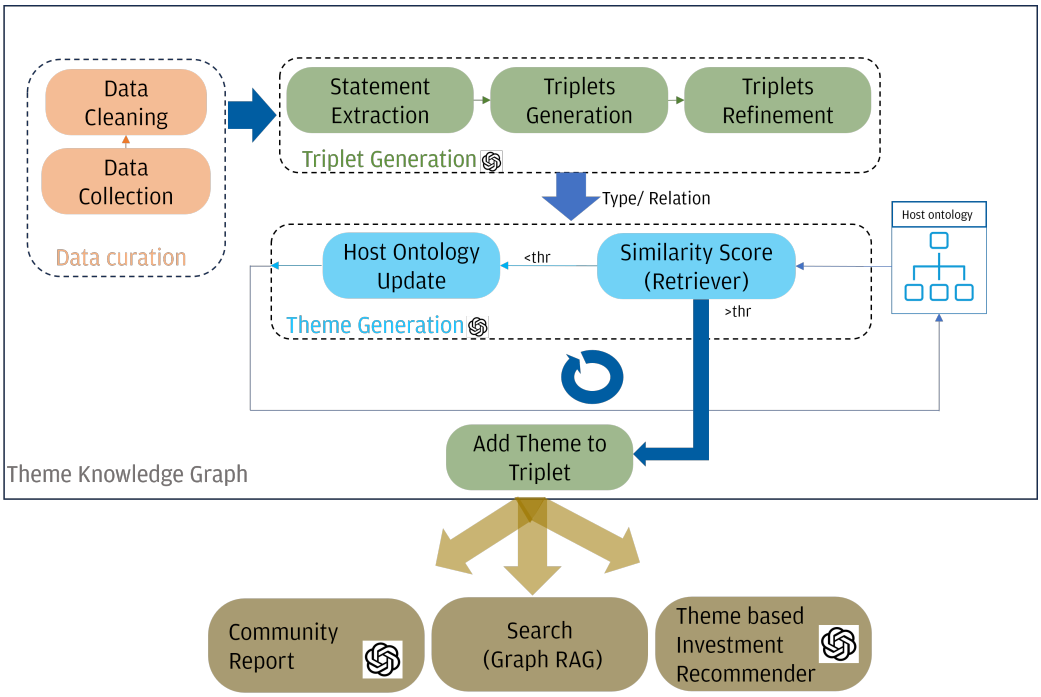

Figure 3: Theme pipeline

B.1 Theme Pipeline

Figure 3 illustrates the theme based search pipeline, which comprises three key steps:

- Data Curation
- Triplet Generation
- Theme Generation

Once the themes are integrated with the triplets, they are indexed using a Retriever system, facilitating efficient investor searches.

Figure 4 illustrates the process of theme generation through taxonomy mapping, which involves the following steps:

- Depth Search Execution: A depth search is conducted on the taxonomy,
- Query Processing where the resulting knowledge graph (KG) relations and node types are submitted as queries to the Retriever host taxonomy
- Similarity Threshold Application (A similarity threshold of 60% is established)
- Taxonomy Expansion (Node types and relations falling below the 60% similarity threshold are collected to form a new taxonomy, which is subsequently integrated into the host taxonomy).

C Knowledge Graph Construction

This appendix contains detailed information about the prompts and hyper-parameters used in the training process.

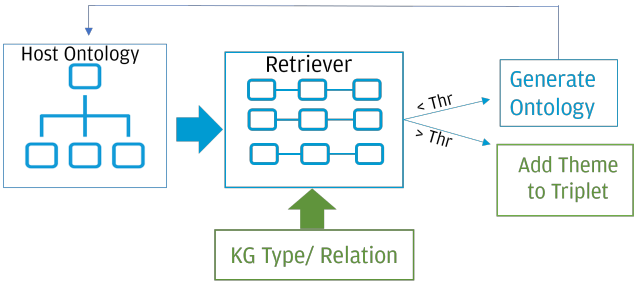

Figure 4: Theme generation through taxonomy mapping

Overview:
You are an expert in economic policy and financial markets and how they affect investment strategies.Your task is to extract key statements from financial commentaries and research provided by Trading and Sales in the credit and macro asset classes following provided guideline.
We are looking for complete, true facts with full context, individualized results, and traceable data.

Business Context:
- Macro Asset Class: Includes, but is not limited to, FX, FX options, rates, swaps, swaptions, government bonds, commodities, and commodity options. It also encompasses emerging market instruments such as eurobonds, eurodollars, and locally issued hard-currency bonds.
- Credit Asset Class: Covers corporate bonds, index options, and CDS instruments.
- Purpose: The extracted information is intended to enhance commentaries accompanying axe-lists distributed by Trading desks directly to clients.

Guideline:
- Self-Contained Statements: Ensure each statement can be understood independently, without external context. If available, attach keywords like country name or date as headers at the beginning of the statement.
- Pronoun Usage: Avoid using ”he” or ”she”; replace with appropriate pronouns where possible.
- Focus Areas: Concentrate solely on market outlooks, commentaries, and updates.
- Exclusions: Disregard non-market or non-economic updates, trader position notes, RFQs, axe tables, price tables, trade-status updates, conference call announcements, non-English text, and emojis.
- Inclusions: Incorporate statements with direct or predicted market or economic impacts, including indicators (e.g., swaps, derivatives, tariffs) and names (e.g., countries, officials, companies).

Review Process:
- Comprehensive Coverage: Ensure all impactful information is included.
- Completeness: Add any missing information to the current list without repetition.
- Contextual Support: Verify that each statement is fully supported by the provided context, avoiding generation of unsupported or hallucinated statements.
- Step-by-Step Approach: Follow the guidelines meticulously to ensure accuracy and relevance.

Examples:
{Examples}

Statements:

Figure 5: Statement extraction prompt for GPT-4o model

C.1 GPT-4o MODEL

C.1.1 STATEMENTS EXTRACTION

Figure 5 is designed for the extraction of key statements. The statistical of the results following the prompt are presented in Table 3.

C.1.2 TRIPLET GENERATION

Figure 6 is the outlined prompt used for generating triplets from extracted statements, aimed at constructing a detailed and accurate knowledge graph. The statistical of the results following the

prompt are presented in Table 4.

C.2 MISTRAL MODEL

In our experiment, the Mistral model was fine-tuned using LoRA for two distinct applications of statement extraction and triplet generation. For both use cases, the training configuration involved four layers of "q_proj", "k_proj", "v_proj" and "o_proj" with parameters set to r=16, lora_alpha=32, and lora_dropout=0.05. The data collation for Mistral was managed using DataCollatorForCompletionOnlyLM, which significantly enhanced the efficiency and effectiveness of the training process.

Training was conducted on the AWS cloud platform using the g5.xlarge instance type, which provides 1 GPU and 4 vCPUs (Amazon Web Services, 2025). During the training phase, we observed that the Mistral model becomes less effective when trained on datasets exceeding 30 entries. This quantity of data is sufficient to demonstrate patterns to the model, while larger datasets tend to interfere with the original weights, resulting in suboptimal training outcomes. The time process for this test was just 4 minutes for 3 epochs. Following are the prompts for both use cases.

Prompt for statement extraction:

```
<s> [INST] Your task is to extract statements with financial
impact from the given context.  Please separate each
statement with "$$$".  The extracted statement should
be self contained and be understood separately out of
context.  Use the exact wording from the context as much
as possible.  The output should only include the extracted
statements, without any additional commentary or repetition
of the context.  Avoid generating any new information or
hallucinations.
## Context:content
## Extracted Statements :[/INST] statements</s>
```

Prompt for triplet extraction:

```
<s> [INST] Your task is to extract the triplets of head ,
tail and relation from the given statement.  Please separate
each triplet with "$$$".:
## Statement:content
## Extracted Triplets :[/INST] triplets</s>"
```

C.3 FLAN T5 MODEL

In this experiment, the Flan T5 model was employed for triplet generation. The training setup included two layers: ["q", "v"]. The LoRA configuration mirrored that of the Mistral model.

Notably, while the Mistral model performed effectively with smaller datasets, the Flan T5 model required a minimum of 2,000 training entries to achieve optimal training outcomes. Training was conducted on similar machine in statement extraction (Amazon Web Services, 2025). The time process for this test was almost 56 minutes for 2 epochs. Following is the prompt for the T5 model.

```
"Extract triplets of head, tail and relation from the
following statement.Please separate each triplet with $$$"
```

D TAXONOMY CONSTRUCTION

This appendix details the taxonomy builder and taxonomy merging

D.1 TAXONOMY BUILDER

The taxonomy builder is configured using GPT-4o. Figure 7 is the prompt for the relation taxonomy builder.

Topic	Question
PREDICT, MARKET MOVEMENT	What are the expectations for US stock futures at the opening in August 2022?
PREDICT POLICY	What was the likelihood of a 50bps rate hike by the central bank in Feb. 2022?
COMMUNICATION, POLICY STRATEGY	What did Fed Chair Powell express about the Fed's pledge to keep monetary policy accommodative in Jan. 2021?

Table 12: Samples of labeled questions by Themesgenerated using LLM (public data)

D.2 TAXONOMY MERGING

Our approach to merging the two taxonomies proceeds in two phases:

In the first phase, we begin with normalization, which eliminates redundancies in vocabulary, for example plural forms (investment vs. investments) and synonyms (AI vs. Artificial Intelligence); we then compare the taxonomies at a higher level by abstracting their paths, temporarily setting aside leaf nodes so that the broader structures can be aligned. The similarity threshold plays a central role in this phase. When parent paths exceed the threshold, we treat them as equivalent; when they fall below the threshold, we either preserve them as distinct categories or seek alignment at a more abstract level.

In the second phase, we bring the parent nodes together and examine their child nodes. If the children exceed the threshold, they are merged, while diverse children below the threshold are retained side-by-side to preserve nuance. By recursively applying this principle down the hierarchy, we arrive at a merged taxonomy that reconciles overlap, preserves necessary distinctions, and maintains semantic richness.

E THEMATIC RETRIEVER EVALUATION

This appendix details the data processing and theme generation methodology used in our study, leveraging publicly available data to ensure reproducibility.

- **Data Collection and Filtering**:
Liang et al. Liang et al. (2024) provided a dataset comprising approximately 15,000 finance-biased statements. We utilized this dataset for our reproducibility test. Initially, we filtered semi-duplicate statements using MinHash (thr=0.4) and Jaccard similarity techniques Yan et al. (2017), resulting in a refined dataset of approximately 3483 unique statements.

- **Triplet Extraction and taxonomy Generation**:
From the filtered statements, we extracted approximately 5176 triplets. The taxonomy generation process was conducted using a large language model (LLM) without human intervention. Similar process explained in 4.1 we generated 80 questions based on the provided themes for retriever comparison. Table 12 is a sample of the questions.

We conducted a similar retriever test available in table 9 on the themed statements. The results are summarized in Table 13, demonstrating that GraphRAG leads precision and F-score.

F ANNOTATION GUIDELINE

In our firm, we avoid duplicating tasks to maintain cost and time efficiency, thanks to our highly trained annotation team specializing in finance data. For each annotation task, we assemble a team of 2-3 members, guided by a team leader.

Following is the workflow of our annotation task.

1- We provide a detailed description of the task, including examples, to explain the annotation goal.

	RAG			GraphRAG		
Topic	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Predict	**0.26**	**0.85**	**0.4**	0.20	0.70	0.31
Market Movement	0.25	0.9	0.39	**0.27**	**0.95**	**0.42**
Performance	0.23	1.0	0.37	0.23	1.0	0.37
Causal Factor	0.21	0.83	0.34	**0.31**	**0.97**	**0.47**
Assessment	0.25	0.95	0.4	**0.31**	**0.95**	**0.47**
Policy	**0.26**	**1.0**	**0.41**	0.24	0.95	0.38
Strategy	0.54	0.88	0.67	**0.57**	**0.89**	**0.69**
Communication	0.40	1.0	0.57	0.40	1.00	0.57
Average	0.24	0.92	0.38	**0.26**	0.92	**0.40**

Table 13: Comparison of Precision, Recall, and F-score between RAG and GraphRAG across different topics

2- Each annotator works through 10-50 selective samples, depending on the complexity of the task (30 samples in our case and we selected the samples from all the available themes in our data for annotation).

3- We hold review meetings to go over annotated samples and discuss complexities and disagreements. During this stage, we update the task description to enhance clarity and reduce variability among annotators.

4- Annotators individually complete their assigned tasks and submit that to a final annotator review.

5- A final annotation review is performed by a domain expert to ensure quality of annotations and resolve conflicts.

Overview:
You are an expert in economic policy and financial markets, tasked with reading the provided statements to generate a detailed and accurate knowledge graph that aids investors.
- Combine Information: Integrate all provided statements into a cohesive relational graph.
- Simplicity and Clarity: Ensure the knowledge graph is straightforward and accessible to a broad audience within the bank.
- Loyalty: Do not introduce information not explicitly mentioned in the text.
- Process: Analyze the text statement by statement, following the process step by step.

Guidelines
Node Types:
- Use consistent and concise terminology.
- Identify financial entities and those impacting economic or market outlooks (e.g., Economic Indicator, Financial Product, Currency, FX Pair, Derivative, Bond, Indices, Country, Economic Policy, Company, Organization, Financial Market, Market Sector).
- Resolve node types where applicable (e.g., Financial Indicator = Economic Indicator, Institution = Organization, Policy = Economic Policy, Currency Pair = FX Pair). - Avoid types with numeric values such as dates, maturities, percentages, volumes, or FX rates.
- Exclude non-informative words (e.g., total, overall, might).
- Use general terms that can encompass multiple values consistently (e.g., ”currency” or ”FX Pair”).

Node Values:
- Use concise identifiers with no more than two words.
- Utilize names or human-readable identifiers found in the text.
- Exclude information about specific JP offers or names of JP Morgan researchers or traders.
- Consider individual bond tickers, FX pairs (e.g., USD/EUR), currencies, curves, indices, and options.
- Include financial indicators of the market environment (e.g., GDP, inflation, employment rates, tax, non-farm).
- Include financial indicators of sentiment and intent (e.g., hawkish, dovish, mean reversion, drop, increase, decrease, saturation, bankruptcy).
- Exclude non-informative words (e.g., total, overall, might).
- Use basic labels for node types (e.g., ”person” for individuals).
- Prefer general terms over specific ones (e.g., ”Bank” instead of ”Bank of US”, ”Trump” instead of ”Trump 2.0” or ”Trump presidency” or ”Trump campaign”, ”Government” instead of ”Government Body”, ”Trading” instead of ”Overall Trading”).
- Use lemmatized and singular node forms (e.g., ’politic’ instead of ’politics’, ’earn’ instead of ’earning’).
- Avoid numeric values such as dates, maturities, percentages, volumes, or FX rates or date-related indicators (e.g., ”Treasury Securities” instead of ”1s/2s/5s”, ”Swap Spread” instead of ”10Y Swap Spread”).

Example for Node Value:
- ”Government 60% tariff on China imports could lead to around 1% drop on GDP in 2025.”
Nodes: ¡Government, Tariff, GDP, China¿ (Note: 60%, 1%, and 2025 cannot be nodes as they are date and percent values).
- ”Trump tax cuts would create a windfall for the top 5% of income earners.”
Nodes: ¡Trump, Tax cut, Income¿ (Note: 5% cannot be a node as it is a percent value).
- ”Migration from Haiti to the United States will boost federal revenues and lower deficits.”
Nodes: ¡Migration, Haiti, United States, Federal Revenue, Deficit¿
- ”USD, equities, crude, and bond yields were higher, with Asian currencies recovering.”
Nodes: ¡USD yield, equity yield, crude yield, bond yield, Asian currency¿

Relation:
- Explain how the head node and tail node are related. The relation should be informative in the context of finance and economy.
- Relation should be factual and evidenced by the text but can be implied by context.
- Use descriptive terms with specific terms that convey clear information and summarize the relationship (e.g., EFFECTS_NEGATIVELY, POSITIVE_SENTIMENT, NEGATIVE_OUTLOOK).
- Avoid to use numeric relations or non descriptive relations such as (BE, CAN, SEE,..) - Ensure consistency and generality.
- Ensure consistency and generality.
Use timeless and lemmatized relation (e.g., ”PRESIDENT” instead of ”BECAME_PRESIDENT”, ”ANNOUNCE” instead of ”ANNOUNCES”, ”WIDEN” instead of ”WIDENED”).

Relation_type:
- Use concise identifiers with no more than two words. (e.g. ”Statement”:”he new policy may increase the market activity”, ”Relation”:”INCREASE”, ”Relation_type”:”PREDICT”).
Examples
Be precise in selecting the relation_type. The goal of Relation_type is to help user to find out the theme of statement.

Coreference Resolution:
- Use the most complete identifier for entities mentioned multiple times with different names or pronouns.

Example for Coreference:
- ”Tariff on import impacts GDP and inflation.”
Relation 1: ¡Tariff, GDP, IMPACT¿. Relation 2: ¡Tariff, Inflation, IMPACT¿

Review Process:
Ensure all important information with financial impact is considered. Add any missing information to the current graph. Do not repeat yourself. JSON records should be unique.
format_instructions
Example for Graph Nodes:
examples

Statements: text

Figure 6: Triplet generation prompt for GPT-4o model

Overview:
You are an expert in the finance domain. You have been provided with a list of keywords including keyword and its theme, separated by commas, extracted from a finance chat call or finance announcement.
Your task is to construct an taxonomy tree from these keywords, with 2-4 hierarchical levels, formatted in JSON.
The goal is to map each keyword to a grandparent node in the taxonomy based on its theme (e.g., (DECREASE, PERFORMANCE) indicates a past occurrence, while (DECREASE, PREDICT) indicates a future prediction).

Guidelines
1- Inclusion: The taxonomy must include all provided keywords and should not introduce any additional keywords.
2- Uniqueness: Ensure there are no duplicate patterns within the taxonomy.
3- Lemmatization: Use lemmatized and singular forms for nodes (e.g. 'APPROVE' insead of 'APPROVES' or 'DECREASE' instead of 'DECREASING').
4- Capitalization: Use Upper case wording (e.g. INCREASE).
5- Exclusions: Avoid using terms like TAXONOMY , FINANCIALTAXONOMY, or any similar combinations as nodes in the taxonomy.
6- Negative Terms: Do not use terms like No, None, Not, or any similar words as nodes in the taxonomy.
7- First Layer Diversity: The taxonomy is set to have multiple nodes in the first layer, as the client prefers an taxonomy with more than one node at the top level.
Here are some example topics for the first layer of taxonomy nodes:
- PREDICT - STRATEGY - COMMUNICATION - MARKET - PERFORMANCE - POLICY - ASSESSMENT - CAUSAL_FACTOR

Review Process After constructing the taxonomy, review it thoroughly. Read through the keywords one by one and ensure all are included in the final taxonomy. The final taxonomy should encompass all the provided keywords.

Example
examples

Figure 7: Relation taxonomy builder