# Contents

# 1 Conceptual Framework

This chapter specifies the conceptual framework for a graph-based data model that represents the causal factors influencing the financial performance of S&P 500 companies. Some general computational approaches for solving the problems stated in Chapter 1 are also provided. Section 1.1 starts with an overview of the primary objectives of the data model, followed by a brief discussion of causal factor representation in Section 1.2. In Section 1.3, a detailed description of the data model and a discussion of its main characteristics are presented.

## 1.1 Overview and Objectives

As mentioned in Chapter 1, the primary motivation for this thesis is to create a text mining tool that mines financial reports to aid investment research. In practice, an important area of focus in investment research concerns companies' business models and underlying industry trends. Some specific questions that investment analysts need to address include:

- What factors affect a particular company's financial performance?
- How do these factors generally change over time? Are there any recognizable patterns?
- Which are the most closely related and comparable peers of a given company?
- Given certain macro-level events, e.g., geopolitical conflicts, food price inflation, interest rate hikes, etc., which companies are most likely to be affected?

For the purpose of this thesis, ***causal factors*** are defined as a set of specific events, a general economic phenomenon, or a category of business activities, etc., which impact the financial performance of a company or a group of companies for a defined period of time. These causal factors necessarily cover a wide range in scope. They are allowed to be as specific or as general as possible. The intention of the flexibility in this definition is to enable capturing not only the immediate, direct drivers of key performance indicators

(KPIs), but also general shifts in macro and sector trends that affect many companies over the long run. Examples of causal factors include: a pandemic such as COVID-19, wage increases, headcount changes, growth in paid subscriber base, etc..

Understanding these causal factors helps investors forecast the future performance of companies under different scenarios to evaluate their intrinsic worth. Comparing a company's relative value with its peers in the same or related sectors can lead to investment opportunities such as the identification of undervalued companies and the early discovery of emerging themes and trends.

Answers to most of these questions can be found in financial reports. However, they do not exist in a readily available form that can be directly retrieved via keyword search. These answers need to be synthesized through incorporation of relevant information, abstraction and summarization. When designing a system that can explicitly address these questions by mining information from financial reports, we gather inspiration from observing financial analysts' workflow and attempt to incorporate as much expert knowledge as possible in it.

The **first observation** is that an experienced analyst usually knows where to look for relevant information in the financial reports. Instead of the most naive approach of reading them linearly in their entirety, he or she can quickly locate the specific sections to focus on, based on awareness of the report's structure and prior knowledge. Moreover, within the relevant section, an analyst also pays more attention to the sentences that explicitly express credit attribution in order to extract useful information. For example, a sentence such as *"There was an increase in operating expenses primarily driven by an increase in compensation expenses largely due to increases in headcount."* is what the analyst would be focused on, if he or she is primarily interested in finding out what causal factors affect the *operating expenses*. In this example sentence, the causal explanation associated with the effect of *"increase in operating expenses"* can be attributed to the explicit factors such as *"an increase in compensation expenses"* and *"increases in headcount"*. We would like our system to incorporate these heuristic rules to facilitate the information extraction process.

The **second observation** is that people naturally process information through abstraction and categorization of similar concepts. Once a causal factor that affects the financial performance of a particular company is identified, the analyst is able to naturally associate it with similar causal factors of related companies. As they accumulate a wealth of associative knowledge throughout the years, they develop a sense about which companies are affected by certain causal factors and in what ways. Therefore, we need a representation of these causal factors which enables our system to perform similar tasks of abstraction and association. In other words, our system needs to be able to cluster these causal factors into meaningful groups that facilitate pattern discovery.

The **third observation** is the dynamic nature of and the interwoven complexity in the

patterns of company-factor relations. We know from experience that some causal factors are company-specific, while others are generic and shared among many companies at the macro-level. In addition, some factors might be related to a one-off event, yet other times they could be recurring or cyclical in nature. Furthermore, it is also essential that the data model is capable of adapting to new patterns. The causal factors evolve with time, hence, the data model needs to be updated incrementally as new information becomes available.

To derive such a data model with these specific requirements, we need a two-staged approach. Firstly, all the causal factors are to be extracted from the raw texts of individual companies' financial reports. Next, the extracted factors are further processed via clustering algorithms. In addition, the data model should be accessible and queryable by users to retrieve data and output visualizations. Figure 1.1 illustrates the envisioned data processing pipeline to achieve this purpose.

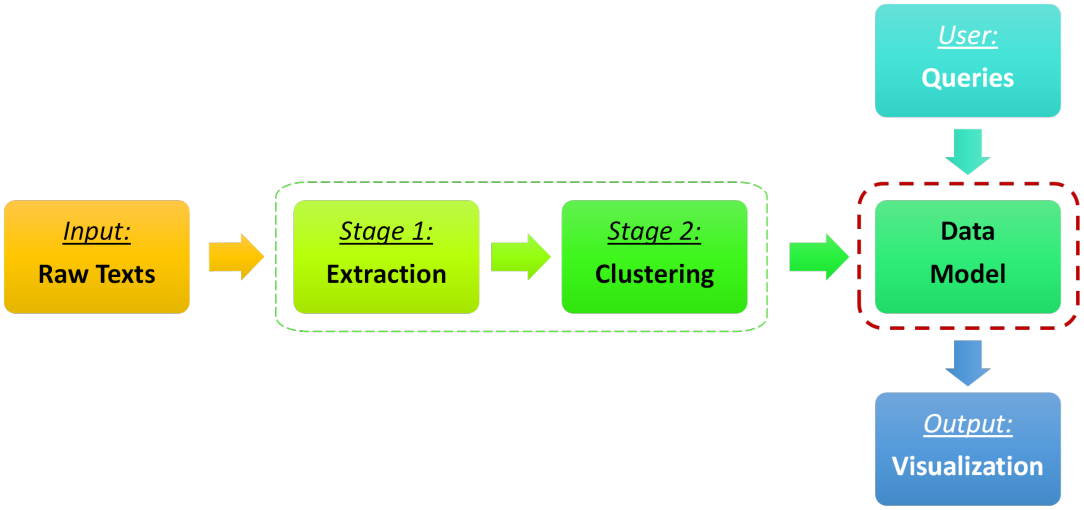


Figure 1.1: Illustration of Data Processing Pipeline

## 1.2 Definitions and Representations

In this section, we define *causal sentences* and *causal factors* in the context of our data model. We first establish a common understanding of causal sentences' structure patterns in terms of *cause* and *effect chunks*. We also introduce the concept of *topics* in relation to causal sentences in financial reports. Finally, we define *causal factors* and discuss their representation.

### 1.2.1 Causal Sentences

In the data model, a ***causal sentence*** is defined as a sentence that contains three components: a cause (C), an effect (E) and an explicit causal connector (CC). A cause is a text fragment within a sentence that describes an event or phenomenon that claims to cause another event or phenomenon, i.e., the effect. These cause and effect text fragments can be a word, a phrase or a clause.

A causal connector is a linguistic signal of causality and it can appear in the form of a verb (e.g., E *driven* by C), a prepositional phrase (e.g., E *due to* C), a conjunction (e.g., E *because* C), etc. It serves as a marker to segment a causal sentence into chunks of text fragments corresponding to cause and effect. For example, the causal sentence, *"The increase in operating expenses was primarily driven by an increase in sales and marketing cost."*, can be represented as: E = *"the increase in operating expenses"*, CC = *"driven by"*, C = *"an increase in sales and marketing cost"*. Any causal sentence can be represented as one of two patterns of cause-effect chunks linked by causal connectors, as illustrated by Figure 1.2 (a) and (b).

Depending on the causal connector's part-of-speech function, as well as the complexity of the corresponding cause and effect chunks, the causal sentence can have a simple, compound or complex sentence structure. For example, a causal sentence such as *"Growth for our direct response advertising products was primarily driven by increased advertiser spending as well as improvements to ad formats and delivery."*, contains multiple causes, C1 =*"increased advertiser spending"* and C2 =*"improvements to ad formats and delivery"*, attributed to one single effect, E = *"Growth for our direct response advertising products"*. In this case, the data model treats C1 and C2 as one combined cause chunk. See illustration in Figure 1.2 (c).

As another example, in the sentence *"There was an increase in operating expenses primarily driven by an increase in compensation expenses largely due to increases in headcount"*, there is a causal chain with multiple causal connectors (*"driven by"*, *"due to"*). In theory, this causal sentence can be broken down into two subsets of cause and effect chunks: 1) E1 = *an increase in operating expenses*, C1 = *an increase in compensation expenses*; 2) E2 = *an increase in compensation expenses*, C2 = *increases in headcount*, where E2 = C1. In this case, the data model treats C1 and C2 as a combined cause chunk, as illustrated in Figure 1.2 (d).

In the context of financial reports, management explains about a company's performance in a formal business language that is logical, concise and unambiguous. Causal sentences typically have an effect chunk referring to a change in one particular financial metric, for example, growth in *revenue*, decline in *gross profit*, improvement in *earnings before interest, tax, depreciation and amortization (EBITDA) margin*, decrease in *net income*, etc. Accordingly, a ***topic*** is defined as a financial metric term that is expressed in an effect chunk of a causal sentence. Specifically, the *topics* referred to in the above

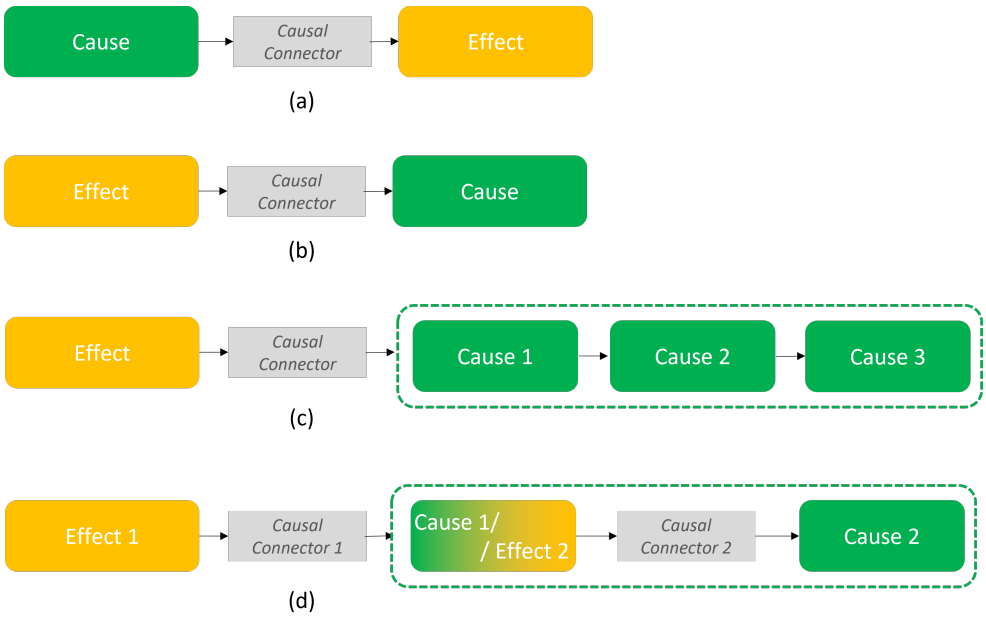


Figure 1.2: Illustration of typical patterns in causal sentences. (a) and (b) are patterns of simple causal sentences involving only one cause and one effect. (c) represents a compound causal sentence with one effect and multiple causes. (d) represents a compound causal sentence with two causal connectors. In both cases (c) and (d), the data model treats the complex causes as a combined chunk.

examples are *revenue, gross profit, EBITDA margin* and *net income*.

### 1.2.2 Causal Factors

The cause chunk in a causal sentence typically expresses the explanation of factors that affect a business' financial performance. In our data model, these factors are referred to as ***causal factors*** (see definition in Section 1.1). Theoretically, there are many possible ways of representing these causal factors.

The most naive way is to treat each cause chunk as one cause factor and store the original text fragments in a database which can be searched via key words. To compare similarities amongst these factors, a bag-of-words or TFIDF approach could be applied. However, the key limitations of this approach is that it ignores the structure and semantic information embedded in text fragments and the weight of each term only reflects the term frequency rather than the functional role in a sentence. In addition, the size of vocabulary as the dimension of the vector representation resulting in sparse data in a high-dimensional space, which is computationally inefficient to deal with.

Another approach is to represent these factors as events with a template specifying a

set of pre-defined features, such as time, location, actor, movement, etc. The advantage is to have a controllable number of features. However, this slot-filling approach requires a lot of expert input in order to manually handcraft. The factors could be diverse and therefore difficult to design a template that fits all scenarios.

Taking inspiration from both of the above mentioned methods, we adopt the following approach: a cause chunk is represented as a directed graph, where the vertices are the noun phrases in the text fragment and the directed edges represents the connecting texts that are in between those noun phrases. The connecting text could be a verb, a conjunction, part of a preposition phrase, etc..

In this model, the noun phrases are treated as carriers of concepts and the connection texts as indication of relationships between these concepts. These noun phrases are further clustered into groups based on semantic similarities. In other words, we group synonyms or phrases representing similar concepts into the same cluster. Ideally, a concept can be identified from each cluster of noun phrases. These clusters can also be considered as equivalent to the slots in a template, except these features are automatically generated based on unsupervised learning rather than manually defined by expert beforehand.

Replacing each noun phrase with its corresponding concept cluster that it belongs to, each cause chunk can therefore be represented as a sequence of connected concept clusters in the direct graph. Instead of using each unique noun phrase as a feature, we use these concept clusters as feature space, thus effectively reducing the dimensionality of feature space to represent these factors.

Alternatively, we could use a language model such as BERT to obtain the contextualized embeddings for these text fragments, however, this black-box approach is not interpretable.

## 1.3 Data Model

In this section, we describe our data model in details. We first establish the data model as a heterogenous graph, then elaborate on the specification for each node type and how the data model is built step by step. We also discuss the key characteristics of the data model.

### 1.3.1 Node and Edges

In essence, our data model is a heterogeneous graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with five distinct node types: *Company*, *Document*, *CausalSentence*, *NounPhrase*, and *Concept*.

$$\mathbf{V} = \mathbf{V}_{company} \cup \mathbf{V}_{document} \cup \mathbf{V}_{causalSentence} \cup \mathbf{V}_{nounPhrase} \cup \mathbf{V}_{concept}$$

There exists a natural order of hierarchy among these nodes, which is indicated by the four distinct types of directed edges:

$$\mathbf{E} = \mathbf{E}_1 \cup \mathbf{E}_2 \cup \mathbf{E}_3 \cup \mathbf{E}_4$$

where

$$\mathbf{E}_1 \subseteq \mathbf{V}_{company} \times \mathbf{V}_{document}$$
$$\mathbf{E}_2 \subseteq \mathbf{V}_{document} \times \mathbf{V}_{causalSentence}$$
$$\mathbf{E}_3 \subseteq \mathbf{V}_{causalSentence} \times \mathbf{V}_{nounPhrase}$$
$$\mathbf{E}_4 \subseteq \mathbf{V}_{nounPhrase} \times \mathbf{V}_{concept}$$

We now elaborate on the specification for each node type and the associated edges. We use (...) to denote *nodes*, $-[...] \rightarrow$ to denote *edges* and {...} to denote edge properties.

**Company Nodes:** Company nodes ($\mathbf{V}_{company}$) represent a set of companies, $\mathcal{C}$, where $\mathbf{V}_{company} \subseteq \mathcal{C}$. Each company node is identifiable by a unique ticker. Additional node properties include the company's full name and its sector classification. For example, Alphabet Inc. is represented as a company node with the ticker *GOOGL*, company name *Alphabet Inc.* and sector classification *Communication Services*.

**Document Nodes:** Document nodes ($\mathbf{V}_{document}$) represent a set of financial documents, $\mathcal{D}$, where $\mathbf{V}_{document} \subseteq \mathcal{D}$. In our case, the set of financial documents consist of form 10-Qs and 10-Ks (see Section 2.3.1). Each financial document is associated with a particular company identifiable by the same unique ticker, as well as a *timestamp* corresponding to the document's reporting period.

**Directed Edge from Company Nodes to Document Nodes:** There is a one-to-many relationship from the company set to the document set. Each company can be associated with multiple documents over an extended period of time, whereas each document can only be associated with one company. The company-document relationship is represented by a directed edge, $\mathbf{E}_1 \subseteq \mathbf{V}_{company} \times \mathbf{V}_{document}$, in the graph. For each

edge, the *timestamp* encoding the document's reporting period is specified as an edge property, as illustrated below:

$$(c_i) - [e_1\{timestamp\}] \rightarrow (d_j)$$

where $c_i \in \mathbf{V}_{company}$, $d_j \in \mathbf{V}_{document}$ and $e_1 \in \mathbf{E}_1$.

**CausalSentence Nodes:** CausalSentence nodes ($\mathbf{V}_{causalSentence}$) represent a set of sentences which express causality, $\mathcal{S}$, where $\mathbf{V}_{causalSentence} \subseteq \mathcal{S}$. In our case, the set of causal sentences are extracted from the set of financial documents $\mathcal{D}$ through a mapping:

$$f_{extract-causal} : \mathcal{D} \rightarrow \mathcal{S}$$

where $f_{extract-causal}$ corresponds to the process of extracting causal sentences from the set of financial documents.

As discussed in Section 3.2, causal sentences can be decomposed into cause and effect chunks, and the effect chunks typically describe changes or movements in some financial metrics, such as increase in revenue, decrease in margins, and increase in net income. Therefore each causal sentence can be tagged with a *topic* which refers to the underlying financial metric reflected in the effect chunk. The typical *topics* include the most common financial Key Performance Indicators (KPIs) such as: *revenue (or sale), cost (or expense), income (or profit), earnings before interest and taxt (EBIT), gross margin, net margin, etc.*.

**Directed Edge from Document Nodes to CausalSentence Nodes:** There is also a one-to-many relationship from the document set to the causalSentence set, since each causal sentence is only associated with one document but multiple causal sentences can be extracted from the same document. The document-causalSentence relationship is represented by a directed edge, $\mathbf{E}_2 \subseteq \mathbf{V}_{document} \times \mathbf{V}_{causalSentence}$, in the graph. For each edge, the *topic* identified from the effect chunk of the causal sentence, is listed as an edge property, as illustrated below:

$$(d_j) - [e_2\{topic\}] \rightarrow (s_k)$$

where $d_j \in \mathbf{V}_{document}$, $s_k \in \mathbf{V}_{causalSentence}$ and $e_2 \in \mathbf{E}_2$.

**NounPhrase Nodes:** NounPhrase nodes ($\mathbf{V}_{nounPhrase}$) represent a set of noun phrases, $\mathcal{N}$, where $\mathbf{V}_{nounPhrase} \subseteq \mathcal{N}$. In our case, the set of noun phrases are extracted from the set of causal sentences $\mathcal{S}$ through a mapping:

$$f_{extract-np} : \mathcal{S} \rightarrow \mathcal{N}$$

where $f_{extract-np}$ represents the process of extracting noun phrases from the cause chunks of the causal sentences, as discussed in Section 3.2. For example, the causal sentence *"The increase was primarily driven by <high levels of promotional expense in the first quarter of fiscal 2015, a decrease in SG&A expenses, and lower supply chain costs>."* contains a cause chunk marked in <...>. This cause chunk can be further segmented into the following noun phrases and represented in an ordered list: [1. *high levels*, 2. *promotional expense*, 3. *the first quarter*, 4. *a decrease*, 5. *SG&A expense*, 6. *lower supply chain costs*] The rational for choosing noun phrases, rather than other part-of-speech terms such as verbs, adjectives, etc., is based on the assumption that noun phrases are the most fundamental symbolic representation of concrete objects as well as abstract ideas, thus a natural choice for the basic lexical unit of concepts.

**Directed Edge from CausalSentence Nodes to NounPhrase Nodes:** There is a many-to-many relationship between the causal sentence set and the noun phrase set. A causal sentence can contain multiple noun phrases and a noun phrase can be contained in multiple causal sentences. In addition, the order of the sequence that the noun phrase appear in a causal sentence is also important and must be encoded in the respective relationship. For the same example above, the edge connecting the noun phrase node *high levels* to the sentence node should an *order* value of 1; the edge associated with *promotional expense* has an *order* value of 2, etc. and so on for the rest of the nouns in the ordered list.

Therefore, the relationships from the causal sentence to each of its constituent noun phrases are represented as directed edges, $\mathbf{E}_3 \subseteq \mathbf{V}_{causalSentence} \times \mathbf{V}_{nounPhrase}$. Each edge has a property that corresponds to the *order* of the noun phrase's appearance in the cause chunk of the causal sentence, as illustrated below:

$$(s_k) - [e_3\{order\}] \rightarrow (n_p)$$

where $s_k \in \mathbf{V}_{causalSentence}$, $n_p \in \mathbf{V}_{nounPhrase}$ and $e_3 \in \mathbf{E}_3$.

**Concept Nodes:** Concept nodes ($\mathbf{V}_{concept}$) represent a set of concepts, $\mathcal{CC}$, where $\mathbf{V}_{concept} \subseteq \mathcal{CC}$. A *concept* is defined as an abstract representation of a group of semantically similar noun phrases. In this case, the set of concepts are formed from partitioning the set of noun phrases into a set of clusters, with each cluster representing a concept

$$f_{cluster-np} : \mathcal{N} \rightarrow \mathcal{CC}$$

where $f_{cluster-np}$ represents the hard clustering process of assigning each noun phrase to one particular concept. For example, noun phrases such as *Europe, EU, Middle East, Africa, Asia Pacific, North Americas*, etc. are expected to form a cluster, from which the concept of *continent or region* can be identified.

**Directed Edge from NounPhrase Nodes to Concept Nodes:** There is a many-to-one relationship between the noun phrase set and the concept set. This results directly from the condition of hard clustering: each concept consists of multiple noun phrases, however, each noun phrase can only be clustered into one concept. The relationships between the noun phrase sets and the concept sets are represented by directed edges, $\mathbf{E}_4 \subseteq \mathbf{V}_{nounPhrase} \times \mathbf{V}_{concept}$, in the graph:

$$(n_p) - [e_4\{order\}] \rightarrow (cc_q)$$

where $n_p \in \mathbf{V}_{nounPhrase}$, $cc_q \in \mathbf{V}_{concept}$ and $e_4 \in \mathbf{E}_4$. There is a potential to extend this model to soft clustering where the edge can also encode the probability. However, this is beyond the scope of this thesis.

Figure 1.3 provides a graphical overview of the set relationships among the five distinct node types in the heterogenous graph-based data model.

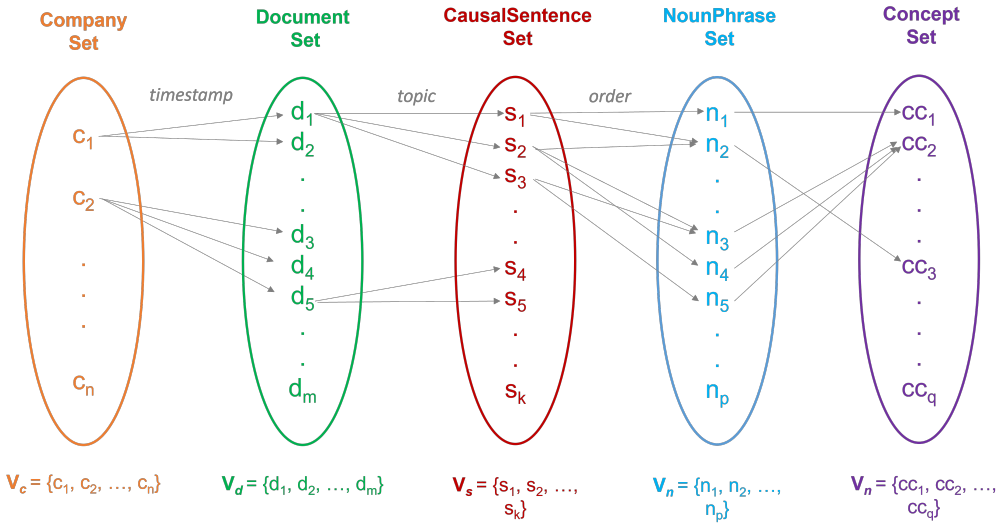


Figure 1.3: Illustration of set relationships among the company set, document set, causalSentence set, nounPhrase set and concept set.

The primary goal of breaking the causal clauses into noun phrases is to transform the data model into a high dimensional latent space, where each specific cause is decomposed into individual features or latent factors. The primary goal of clustering noun phrases into concepts is to detect the hidden patterns and reduce the dimensionality of the sparse latent space. This effectively achieves data compression for generalization and data association in order to facilitate data retrieval. In addition, these concept clusters also enable direct connections to be made between companies which might be affected by implicitly similar factors that are semantically close in the latent space, but not expressed by exactly the same causal clauses at the explicit level.

In summary, our data model has the following characteristics:

- It is a heterogeneous graph with five distinct types of nodes: *Company, Document, CausalSentence, NounPhrase, and ConceptCluster.*
- These nodes follow a natural hierarchical order: *Company* → *Document* → *CausalSentence* → *NounPhrase* → *ConceptCluster*
- Concept nodes can be perceived as the connection points providing the means for similarity measures at different node levels.
- Temporal information included in the graph make it possible to treat it as a dynamic graph, where an evolution of trends can be observed with respect to the passage of time.
- The data model can be incrementally updated, when new financial reports become available.

### 1.3.2 Node Embeddings and Similarity Measures

Having established the basics of the data model, we now move on to define node embeddings and similarity measures between nodes. This paves the foundation for directly addressing the two objectives stated earlier in Section 3.1: the identification of performance drivers and the discovery of similar companies. Node embeddings and similarity measures are defined according to their respective node types.

**NounPhrase Node Embedding:** Since a $NounPhrase$ node represents a noun or a noun phrase, the underlying word embedding is a natural choice for the node embedding. Different types of word embeddings, such as word2vec, GloVe and BERT, have already been discussed in Section 2.2.1. In addition, the various methods for computing phrase represenations from word vectors have also been discussed in Section 2.2.2. A suitable choice for the noun phrase node embeddings in our data model is a static word-level embedding model pre-trained on a relatively large corpora, such as pretrained GloVe embeddings. A simple average operation is opted for obtaining the compositional phrase embeddings from the constituents word embeddings.

$$\mathbf{Embedding}(NounPhrase) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{w_i}$$

where $\mathbf{w_i}$ is the word embedding for the $i^{th}$ token in the $NounPhrase$ and $n$ is the total number of tokens in the $NounPhrase$.

The similarity between a pair of $NounPhrase$ nodes is defined as the cosine similarity

between the node embeddings.

$$\textbf{Similarity}(NounPhrase_A, NounPhrase_B) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}$$

where $\mathbf{A} = \textbf{Embedding}(NounPhrase_A)$, $\mathbf{B} = \textbf{Embedding}(NounPhrase_B)$.

These similarity measures allows the grouping together of noun phrases which are closely related in semantics, while spacing wider apart the ones which are distantly related. These node embeddings are used as input to the clustering algorithm which generate *Concept* nodes.

**Concept Node Embedding:** A *Concept* node is effectively the centroid of the clusters of *NounPhrase* nodes which are directly connected to it in the graph. The node embedding for a *Concept* node is defined as:

$$\textbf{Embedding}(Concept) = \frac{1}{m} \sum_{j=1}^{n} \mathbf{V_j}$$

where $\mathbf{V_j}$ is the node embedding for the $j^{th}$ *NounPhrase* in the immediate neighborhood of the *Concept* node and $m$ is the total number of *NounPhrase* nodes in the neighbourhood.

The similarity between a pair of *Concept* nodes is defined as the cosine similarity between the node embeddings.

$$\textbf{Similarity}(Concept_X, Concept_Y) = \frac{\mathbf{X} \cdot \mathbf{Y}}{\|\mathbf{X}\| \cdot \|\mathbf{Y}\|}$$

where $\mathbf{X} = \textbf{Embedding}(Concept_X)$, $\mathbf{Y} = \textbf{Embedding}(Concept_Y)$.

**CausalSentence Node Embedding:** Each *CausalSentence* node is connected to an ordered set of *NounPhrase* nodes that represent the constituent underlying noun phrases according to their order of appearance in the cause chunk of the sentence. Each *NounPhrase* node is in turn connected to a *Concept* node. Therefore, the node embedding for a *CausalSentence* node can be represented as a list of *Concept* node indices:

$$\textbf{Embedding}(CausalSentence) = [cc_1, cc_2, ..., cc_i, ..., cc_n]$$

where $cc_i$ is the index of the *Concept* node linked to the *NounPhrase* node, which is connected to the *CausalSentence* node with an edge order $= i$; $n$ is the total number of *NounPhrase* nodes connected to the *CausalSentence* node.

The *CausalSentence* node embeddings are allowed to have different lengths according

to how many *NounPhrase* nodes there are in its neighborhood. In other words, the lengths of the node embeddings also encode the amount of information content in the sentences represented by the *CausalSentence* nodes.

The similarity between a pair of *CausalSentence* nodes is measured at two levels: weak similarity and strong similarity. The weak similarity is defined as the Jaccard similarity between the two node embeddings of the *CausalSentence* nodes:

$$\textbf{Similarity}(CausalSentence_A, CausalSentence_B) = \frac{\|\mathbf{A} \cap \mathbf{B}\|}{\|\mathbf{A} \cup \mathbf{B}\|}$$

where $\mathbf{A} = \textbf{Embedding}(CausalSentence_A)$, $\mathbf{B} = \textbf{Embedding}(CausalSentence_B)$.

Only if the weak similarity between two *CausalSentence* nodes are above a certain threshold, then the strong similarity is defined. The strong similarity is used as a more precise measure of how similar the pair of nodes are when they are already considered somewhat similar by the weak similarity measure. If the pair of nodes has a weak similarity below the threshold, for example, when the pair of *CausalSentence* nodes do not share any *Concept* nodes in common, there is no practical need to further compute the strong similarity score between them.

There are various methods to define the strong similarity, such as Levenshtein distance [1] or Sorenseon-Dice coefficient [1] based on the *CausalSentence* node embeddings; alternatively, a cosine similarity between the sentence embeddings of the underlying text can be obtained from a pretrained model such as Sentence-BERT [2]. The precision required for the strong similarity measure depends on the end application and how much computational resources are available in practice. The exploration in this regard is left to future studies.

For the purpose of this thesis, only the weak similarity measure is applied, as there is no practical need to compare two individual sentences in the end application.

**Document Node Embedding:** Each *Document* node is connected to a set of *CausalSentence* nodes via edges labelled with *topics*, which are financial KPIs explained by the causal sentences. Accordingly, the *Document* node embedding is defined as the collection of different sets of the *CausalSentence* node embeddings, where *CausalSentence* notes in each set correspond to a distinct *topic*.

$$\textbf{Embedding}(Document) = \{\mathbf{D_t} \mid t \in topics(Document)\}$$

where

$$\mathbf{D_t} = \{\textbf{Embedding}(CausalSentence_i) \mid CausalSentence_i \in \mathcal{N}_t(Document)\}$$

and $\mathbf{D_t}$ represents a set of *CausalSentence* node embeddings under a topic $t$, $topics(Document)$ represents the set of *topics* on the outgoing edges of the *Document* node, $\mathcal{N}_t(Document)$ represents the set of $CausalSentence_i$ nodes in the immediate neighborhood of the *Document* node under the topic $t$.

Equivalently, each *Document* node embedding can be expressed in the form of a matrix, where each column is a vector representation of the distribution of the *Concept* nodes under a *topic*. The dimension of the column vector is equal to the total number of unique *Concept* nodes and each entry in the embedding corresponds to the number of counts for each *Concept* node.

$$\textbf{Embedding}(Document) = (d_{i,j}) \in \mathbb{R}^{m \times n}$$

where $m =\mid \mathbf{V}_{Concept} \mid$, $n =\mid \mathbf{E}_2 \mid$.

The similarity measure between two *Document* nodes is defined as the cosine similarities of each pair of corresponding column vectors of the two nodes embedding matrices. The resulting representation is a row vector with each entry representing the similarity measure for each topic.

$$\textbf{Similarity}(Document_A, Document_B) = [\mathbf{s_j}] \in \mathbb{R}^{1 \times n}$$

$$\mathbf{s_j} = \frac{\mathbf{a_j} \cdot \mathbf{b_j}}{\|\mathbf{a_j}\| \cdot \|\mathbf{b_j}\|}$$

where

$$\mathbf{a_j} = \textbf{Embedding}(Document_A)[:, j]$$

and

$$\mathbf{b_j} = \textbf{Embedding}(Document_B)[:, j]$$

**Company Node Embedding:** Each *Company* node is connected to a set of *Document* nodes through timestamped edges. A *Company* node embedding is therefore represented as a tensor, i.e., a stack of *Document* embedding matrices each representing a discrete timestamp. This approach naturally supports generating dynamic *Company* node embeddings based on the specification of *timeperiods* according to the use cases.

$$\textbf{Embedding}(Company_{t_1}^{t_2}) = \sum_{t=t_1}^{t=t_2} \textbf{Embedding}(Document_t)$$

where $Document_t \in \mathcal{N}_t(Company)$, the neighborhood of the *Company* node for the specified period from $t = t_1$ to $t = t_2$.

For a specified time period, the three-dimensional tensor representations of a *Company* node embedding can be effectively transformed into a two-dimensional matrix by ag-

gregating along the temporal axis by summation; the resulting matrix representation is effectively in the same form as a *Document* node embedding as described above.

The similarity measure between two *Company* nodes, $(Company_A)_{t_1}^{t_2}$ for the time period from $t1$ to $t2$ and $(Company_B)_{\tau_1}^{\tau_2}$ for the time period from $\tau_1$ to $\tau_2$, are defined the cosine similarities of each pair of corresponding column vectors of the two nodes embedding matrices:

$$\textbf{Similarity}((Company_A)_{t_1}^{t_2}, (Company_B)_{\tau_1}^{\tau_2}) = [\mathbf{s_j}] \in \mathbb{R}^{1\times n}$$

$$\mathbf{s_j} = \frac{\mathbf{a_j^T} \cdot \mathbf{b_j^\mathcal{T}}}{\|\mathbf{a_j^T}\| \cdot \|\mathbf{b_j^\mathcal{T}}\|}$$

where

$$\mathbf{a_j^T} = \textbf{Embedding}((Company_A)_{t_1}^{t_2})[:, j]$$

and

$$\mathbf{b_j^\mathcal{T}} = \textbf{Embedding}((Company_B)_{\tau_1}^{\tau_2})[:, j]$$

This approach not only allows the optional tracking of the temporal evolution of the *Company* node embeddings at various granularity levels, e.g., quarterly, annually or every five years, but also enables comparison of companies across different periods of time.

# Bibliography

[1] S. Ontanon, "An overview of distance and similarity functions for structured data," *Artificial Intelligence Review*, pp. 5309 – 5351, October 2020.

[2] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019.