[8]/10 [ToDo] 2021 - thesis - 73 pages - 2021.05

DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

Exploring Construction of a Company Domain-Specific Knowledge Graph from Financial Texts Using Hybrid Information Extraction

CHUN-HENG JEN

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Exploring Construction of a Company Domain-Specific Knowledge Graph From Financial Texts Using Hybrid Information Extraction

Chun-Heng Jen

Authors

Chun-Heng Jen <chjen@kth.se>
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden

Examiner

Markus Hidell
KTH Royal Institute of Technology

Supervisor

Peter Sjödin

KTH Royal Institute of Technology

Abstract

Companies do not exist in isolation. They are embedded in structural relationships with each other. Mapping a given company's relationships with other companies in terms of competitors, subsidiaries, suppliers, and customers are key to understanding a company's major risk factors and opportunities. Conventionally, obtaining and staying up to date with this key knowledge was achieved by reading financial news and reports by highly skilled manual labor like a financial analyst. However, with the development of Natural Language Processing (NLP) and graph databases, it is now possible to systematically extract and store structured information from unstructured data sources. The current go-to method to effectively extract information uses supervised machine learning models, which require a large amount of labeled training data. The data labeling process is usually time-consuming and hard to get in a domain-specific area.

This project explores an approach to construct a company domain-specific Knowledge Graph (KG) that contains company-related entities and relationships from the U.S. Securities and Exchange Commission (SEC) 10-K filings by combining a pre-trained general NLP with rule-based patterns in Named Entity Recognition (NER) and Relation Extraction (RE). This approach eliminates the time-consuming data-labeling task in the statistical approach, and by evaluating ten 10-k filings, the model has the overall Recall of 53.6%, Precision of 75.7%, and the F1-score of 62.8%. The result shows it is possible to extract company information using the hybrid methods, which does not require a large amount of labeled training data. However, the project requires the time-consuming process of finding lexical patterns from sentences to extract company-related entities and relationships.

Keywords

Natural Language Processing, Information Extraction, Named Entity Recognition, Relation Extraction, Knowledge Graph

Sammanfattning

Företag existerar inte som isolerade organisationer. De är inbäddade i strukturella relationer med varandra. Att kartlägga ett visst företags relationer med andra företag när det gäller konkurrenter, dotterbolag, leverantörer och kunder är nyckeln till att förstå företagets huvudsakliga riskfaktorer och möjligheter. Det konventionella sättet att hålla sig uppdaterad med denna viktiga kunskap var genom att läsa ekonomiska nyheter och rapporter från högkvalificerad manuell arbetskraft som till exempel en finansanalytiker. Men med utvecklingen av ”Natural Language Processing” (NLP) och grafdatabaser är det nu möjligt att systematiskt extrahera och lagra strukturerad information från ostrukturerade datakällor. Den nuvarande metoden för att effektivt extrahera information använder övervakade maskininlärningsmodeller som kräver en stor mängd märkta träningsdata. Datamärkningsprocessen är vanligtvis tidskrävande och svår att få i ett domänspecifikt område.

Detta projekt utforskar ett tillvägagångssätt för att konstruera en företagsdomänspecifikt ”Knowledge Graph” (KG) som innehåller företagsrelaterade enheter och relationer från SEC 10-K-arkivering genom att kombinera en i förväg tränad allmän NLP med regelbaserade mönster i ”Named Entity Recognition” (NER) och ”Relation Extraction” (RE). Detta tillvägagångssätt eliminerar den tidskrävande datamärkningsuppgiften i det statistiska tillvägagångssättet och genom att utvärdera tio SEC 10-K arkiv har modellen den totala återkallelsen på 53,6 %, precision på 75,7 % och F1-poängen på 62,8 %. Resultatet visar att det är möjligt att extrahera företagsinformation med hybridmetoderna, vilket inte kräver en stor mängd märkta träningsdata. Projektet kräver dock en tidskrävande process för att hitta lexikala mönster från meningar för att extrahera företagsrelaterade enheter och relationer.

Nyckelord

Naturlig språkbehandling, Informationsextraktion, Namngiven Entitetsigenkänning, Relationsextraktion, Kunskapsgraf

Acronyms

KG	Knowledge Graph
NER	Named Entity Recognition
NLP	Natural Language Processing
IE	Information Extraction
RE	Relation Extraction
POS	Part-of-speech
AI	Artificial Intelligence
OIE	Open Information Extraction
SDG	Sustainable Development Goals
CRF	Conditional Random Fields
SEC	U.S. Securities and Exchange Commission
SVM	Support Vector Machines
CRISP-DM	CRoss Industry Standard Process for Data Mining
EDGAR	Electronic Data Gathering, Analysis, and Retrieval system

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Companies do not exist in isolation but are embedded in structural relationships with each other [1]. Mapping a given company's relationships with other companies in terms of competitors, subsidiary, suppliers, customers is key to understand the major risk factors and opportunities for a company. For example, delays or closure to one of the company's major suppliers can negatively impact its production. Similarly, a company might lose market share due to a competitor releasing a product with a superior feature. Aside from a company's point of view, a bank can evaluate the risk of a loan based on a company's relationships. Also, an investment firm can obtain investment insight by having a clear context of a company's relationship with other companies. Thus, a better understanding of companies' structural relationships can lead to improved investment performance and enhance the companies' transparency. This is evidenced by the fact that data vendors have started to provide curated data sets describing company relationships to investment firms.

However, company relationships can be complex, especially as these relationships evolve over time [2]. As a result, these curated data sets from data vendors can be expensive, limited in scope, and valid for a fixed period of time. On the other hand, there is a huge volume of public financial information such as financial news, SEC-filings (10-K), broker reports, and tweets available to extract company relationships. SEC 10-K filings, in particular, is a detailed annual legal filing reported by all US publicly-listed companies. The report includes information such as a company's important business activities, competitors, subsidiaries, financial statements, earnings, and other relevant information about a company.

Conventionally, obtaining this key knowledge and staying up to date with those company relationships are achieved through highly skilled manual labor such as financial analyst and paralegal. However, it is inefficient and costly to have human reading and keep up to date with all the news, filings, and reports. It also lacks a central place for storing and accessing this key knowledge once it is obtained. Therefore, this can lead to repetitive tasks to obtain the same knowledge overtime. As a result, a systematic approach using a machine can decrease time and cost, increase the accessibility and transparency of the key information, and allow employees to work on more high impact tasks. With the evolution in machine learning, natural language processing, and graph database, it is now possible to analyze unstructured data on social media, public conference calls, and legal documents and store information extracted.

This project aims to explore a viable method to systematically extract and store key relationships about companies from the financial text.

1.1 Background

The Oxford dictionary defined knowledge as ”Facts, information, and skills acquired through experience or education” [3]. Human gains knowledge through the process of finding and memorizing information from trusted sources [4]. The same goes for a machine that needs to extract information from a trusted source and store the information in a database. As a result, there are two steps for a machine to systematically obtain and maintain knowledge of facts.

The first step is to extract facts from structured and unstructured text corpora. This is done by using Information Extraction (IE), an area of Natural Language Processing (NLP) research. Multiple subfields of study have been introduced for IE, such as Named Entity Recognition (NER), Relation Extraction (RE), Open Information Extraction (OIE), Event Extraction, and more [5][6].

The second step is to store, maintain, and update the retrieved knowledge. A Knowledge Graph (KG) system was proposed and developed by Metaweb in 2007 [7]. Metaweb’s KG stores the extracted information from Web pages such as Wikipedia in a graph database. In a graph database, the information is stored in a structured way, using a set of triples [8]. A triple consists of 2 entities/subjects/objects, such as people, organization, events, and the relationship between them. By having the KG approach, it becomes easier to create, update, maintain, and delete knowledge and allows advanced applications to be developed. An example of the most known KG was commercially developed in 2012 by Google. Google's Knowledge Graph is a General KG that allows Google to provide users relevant information that is not included in the search words. For example, when one searches Mona Lisa, Google can present information about the artwork's location, the year it is created, the artist who painted it, and more.

Besides Google's storage of general information, a KG can be used to tie specific relationships and specific entities together to create a domain-specific KG. Domain-specific KG can contain more complex entities and specific relationships from domain-specific data, which can be used for diverse cases. For example, domain-specific KG has been used for cybersecurity where the network environment is mapped and visualized to allow a comprehensive view of the network with regards to vulnerabilities when they occur. This enables isolated cyber event management and a big picture analytic to ensure the security of the mapped network [9]. Furthermore, domain-specific KG has also been used to mapped human trafficking, fraud detection, forecasting geopolitical events, and more [10]. Because a domain-specific KG can provide a comprehensive view of the structural relationship between each company, it is an ideal approach to represent company-specific knowledge such as companies' products, services, and companies' relationships with other companies in terms of competitors, subsidiary, and more.

Therefore, this project focuses on exploring how to construct a company domain KG by extracting and storing company-related entities and relationships from the financial text.

1.2 Problem

Currently, there are two challenges in constructing a domain-specific KG. First, although many publicly available IE models have good performance in extracting general information, they cannot achieve the same performance in a domain-specific context. Second, state-of-the-art IE models use a supervised machine learning method, which uses a large amount of labeled training data. For general IE, there are publicly available labeled data for training. However, for domain-specific IE, there tends to have little publicly available labeled data for training. Thus it requires time-consuming data labeling tasks to create a large amount of training data [10].

Additionally, existing research on constructing a company domain KG focus on data from either news and articles in English or financial filings in Chinese [11][12][13] and during the literature research process, this project found there is limited research on building a company domain KG from the SEC 10-K filings, a financial legal filing that is in English.

To solve the above challenges and reduce time and cost in obtaining and updating company relationships, this thesis aims to explore a domain-specific KG approach of extracting and storing company-related entities and relationships from the SEC 10-K filings that does not require time-consuming data labeling tasks.

To summarize, this thesis aims to answer the following two questions:

1. How to construct a domain-specific KG that contains company-related entities and relationships from unstructured text, the SEC 10-K filings?
2. Is it possible to eliminate the time-consuming data-labeling task when constructing a domain-specific KG that can have achieved a certain level of performance?

1.3 Purpose

The purpose of the project is to reduce the time and the cost of reading the SEC 10-K filings and increase the accessibility of the obtained information by exploring the approach of domain-specific KG to extract and store company-related entities and relationships from the SEC 10-K filings, which are unstructured text corpus.

1.4 Goal

The goal of this project is to design and develop a domain-specific KG that can extract company-related entities and relationships from SEC 10-K filings by eliminating the time-consuming data-labeling tasks and form and store the triples.

1.5 Benefits, Ethics and Sustainability

This project can benefit organizations by reducing time and cost in mapping out organizations' relationships with other organizations. This allows organizations to have a better understanding and control of information and mitigate human biases by enhancing the transparency and accessibility of company-related information. Additionally, a company's relationships KG allow the development of financial applications ranging from stock price movement, to industry mapping, and to risk analysis of credits in the financial services industry.

Since all the data used in this project are public, there are no licenses nor privacy concerns. This project is done with a sponsor company and the implementation of the application may be confidential.

United Nations defined 17 Sustainable Development Goals (SDG) in 2015 to encourage nations, industries, and researchers to focus on sustainable growth and provide a better future for the next generation[14]. This project tries to reduce the human cost for the extraction of companies' relationships and to have more accurate information about different companies. Therefore, this project may help the sustainable development of the following two targets in SDG 8 Decent Work and Economic Growth and SDG 9 Industry, Innovation, and Infrastructure:

- One of the SDG targets[14]: "Increase the access of small-scale industrial and other enterprises, in particular in developing countries, to financial services, including affordable credit, and their integration into value chains and markets."
- One of the SDG targets[14]: "Strengthen the capacity of domestic financial institutions to encourage and expand access to banking, insurance, and financial services for all."

1.6 Methodology

Based on Håkansson's paper [15], this project is based on qualitative research with an empirical research method to answer the feasibility of extracting company-related entities and relationships and building a domain-specific KG using the hybrid IE. This solution is based on extensive research on related works on IE and uses a design science approach to build the domain-specific KG. Additionally, the project uses quantitative statistical methods to analyze the performance of the KG and answer the problems stated in Section 1.2.

1.7 Delimitations

The project focuses on exploring the feasibility of using IE and KG to extract and store company-related entities and relationships in terms of competitors, subsidiary, company products from financial text data. Therefore, the developed KG may not have the best performance due to the selected IE methods. Also, the performance of the application in terms of run-time speed and resource efficiency is not analyzed in this project. Furthermore, the performance of the Domain Specific KG in terms of scalability is also not considered in this thesis together with other relationships that could be extracted are not analyzed.

The development risk in terms of development time, data rights, technology usages, and computing power has been considered.

1.8 Outline

The report provides relevant background information about SEC 10-K filings NLP, and KG construction in Chapter 2. In Chapter 3, the framework to solve the problem is presented, and the techniques that were used for this thesis are introduced. Chapter 4 shows how the solution is implemented. Chapter 5 provides an analysis of the results of the proposed solution. Lastly, Chapter 6 concludes by discussing the findings.

Chapter 2

Background and Related Work

In this chapter, a detailed description of the degree project background is presented together with related work. In Section 2.1, the general information about the SEC 10-K filing is described. Section 2.2 describes the general usage of NLP and the rule-based and statistical-based approach to building a NLP. Section 2.3 contains the detailed background of the KG, KG structure, KG's application, different NLP techniques to perform information extraction to extract knowledge from unstructured data, and how to store and maintain the extracted knowledge effectively. Section 2.4 presents related work on how to extract information and build a knowledge graph both in general and domain-specific areas.

2.1 SEC 10-K Filings

This thesis focuses on financial data that are submitted to the SEC. The SEC is an independent federal government agency in the USA. Public companies are legally required to disclose company information that may affect shareholders, financial markets, and the general state of the economy [16]. This information includes, among others, a company's business, market, competitor, financial data, legal proceeding, and executive's compensation, allowing one to understand the company's business better.

Particularly, Item I, the business section in the SEC 10-K filing, contains information about the company business activities such as the event of merger and acquisition, the products or services the company sells, and its competition with other companies.

This is evidenced by the following sentences from Microsoft's latest 10-K filing. ”On October 25, 2018, Microsoft acquired GitHub, Inc. in a $7.5 billion stock transaction.”, ”Microsoft's Intelligent Cloud segment primarily comprises Server products and cloud services, including SQL Server, Windows Server, Visual Studio, System Center, and related CALs, GitHub, and Azure.”, and ”Microsoft's products for software developers compete against offerings from Adobe, IBM, Oracle, and other companies, and also against open-source projects, including Eclipse (sponsored by CA Technologies, IBM, Oracle, and SAP), PHP, and Ruby on Rails”[17]. This project aims to systematically extract the above information.

2.2 Natural Language Processing

NLP is an area of computer science and Artificial Intelligence (AI) that study how to allow computer systems to understand or produce existing human language by processing unstructured textual or audio information [18]. NLP allows applications such as Facebook's sentiment analysis for building user profiles, Google language translation, and Speech recognition in Siri.

NLP has developed into many subareas of study: “Computational Linguistics, Information Extraction (IE), Information Retrieval, Natural Language Understanding and Natural Language Generation” [19]. This thesis focuses on IE which is described in detail in Section 2.3.1.

Most of the NLP methods can be developed using either rule-based or statistical-based methods.

2.2.1 Rule-based

Rule-based methods use linguistic structures such as language grammar, syntax, word pattern, and a specific list of words to allow the machine to understand or produce language. For example, in information extraction, rule-based systems may analyze the sentence structure and use an existing dictionary to identify the noun, subject, and object in the sentence [6]. Another example, sentiment analysis with a rule-based approach may use a list of words and data labeling to classify whether a given word, sentence, or document is in positive, negative, or neutral emotion[20].

The rule-based method can be time-consuming and requires input from linguists as it requires extensive linguistic knowledge to set up handcrafted rules for the rule-based systems to work. However, the rule-based approach can produce a more consistent result since rule-based methods do not depend on the quality of training data sets. As a result, the rule-based method can be an ideal approach in building KG using 10-K filings if there is a lack of training data, and consistency of the output is a requirement.

2.2.2 Statistical-based

NLP development focuses now more on statistical-based methods as it uses the "large amounts of text data" to train a statistical model to learn language grammar, syntax, word pattern, etc.

NLP that uses Statistical based methods usually deploys different machine learning algorithms to derive knowledge/insight from the input text/audio data. Common machine learning algorithms used in NLP are decision tree, Support Vector Machines (SVM), and Conditional Random Fields (CRF) for supervised learning and clustering, and Hidden Markov models for unsupervised learning. The statistical-based approaches can have a higher coverage of cases because their large training data set enables to encompass different scenarios. Therefore, the statistical-based approach is more scalable than the rule-based method since it does not require humans to set up rules. However, the statistical-based method's performance is highly dependent on the quality and quantity of the training data.

Because of the availability of a large amount of general data, the statistical-based method is highly used in the common scenarios.

2.3 Knowledge Graph Construction

Kejriwal wrote, ”a KG is a graph-theoretic representation of human knowledge such that it can be ingested with semantics by a machine” and ”KG is a way to represent 'knowledge' using graphs so that a machine would be able to conduct reasoning and inference over this graph to answer queries ('questions') in some meaningful way.”[10].

In technical terms, KG can be seen as a graph database that stores information in the form of triples where each triple is an assertion of fact [10]. A triple (h, r, t) consists of a head entity h, tail entity t, and the relationship between the two entities r (see Figure 2.3.1).

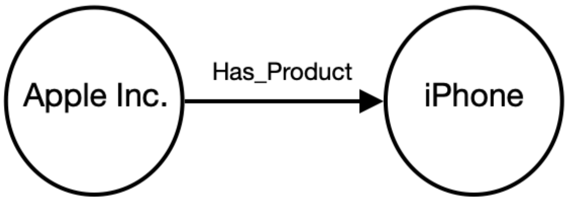
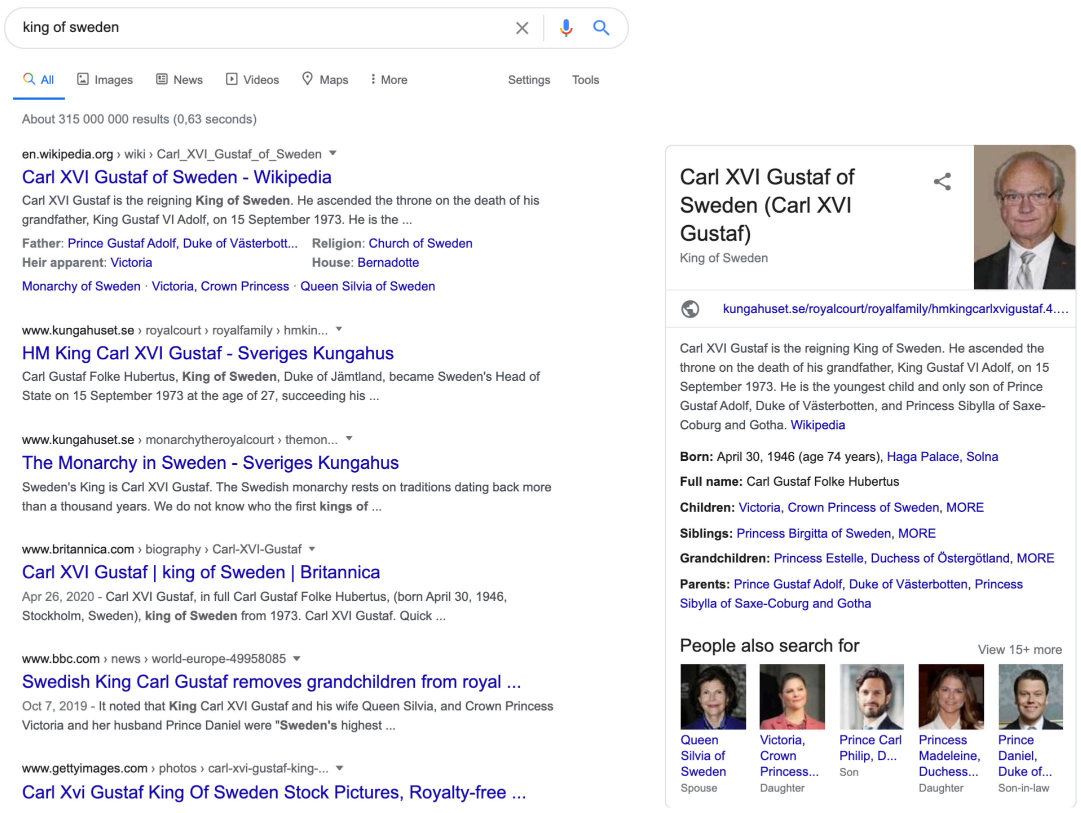


Figure 2.3.1: Example of a Triple (Source: Author)

Because of the structured nature of the triple, computer systems can process the assertions/facts from KG and provide a comprehensive view based on the KG. As a result, by leveraging KG, more sophisticated applications can be developed. For example, the first notable use of a KG is Google Knowledge Graph, which Google incorporates it within its search engine [21]. Using Google Knowledge Graph, Google can provide search results based on the user's string input and provide a richer search result by analyzing entities and relationships in the string input and providing relevant entities and relationships based on the Google Knowledge Graph. As a result, Google can provide a better search experience for its users. For example, one can see in Figure 2.3.2, when one searches King of Sweden, Google search engine can present information about the current king of Sweden, Carl XVL Gustaf of Sweden, as well as his birthday, full name, siblings, and children on the right side of the page. This is achieved because Google Knowledge Graph contains a triple of (h: Carl XVL Gustaf of Sweden, r: is, t: King of Sweden).

Besides, Google's Knowledge Graph, where it contains general facts such as notable people, time, and event, a KG can also contain more domain-specific knowledge. For example, a company domain knowledge graph that stores information about different companies' products, competitors, suppliers, buyers, and shareholders can allow one to have a comprehensive picture of their operations and competition. Moreover, KG can also provide insightful information when a set of companies' information is combined to provide a view of an industry.

As the company-related entities and relationships with each other can be found in the SEC 10-K filings, this thesis focuses on exploring using IE to extract company information from 10-K filings and represent the extracted information in the form of triples to build a company domain KG. The following subsections describe the



king of sweden

All · Images · News · Videos · Maps · More · Settings · Tools

About 315 000 000 results (0,63 seconds)

en.wikipedia.org › wiki › Carl_XVI_Gustaf_of_Sweden
Carl XVI Gustaf of Sweden - Wikipedia
Carl XVI Gustaf is the reigning **King of Sweden**. He ascended the throne on the death of his grandfather, King Gustaf VI Adolf, on 15 September 1973. He is the ...
Father: Prince Gustaf Adolf, Duke of Västerbott... **Religion**: Church of Sweden
Heir apparent: Victoria **House**: Bernadotte
Monarchy of Sweden · Victoria, Crown Princess · Queen Silvia of Sweden

www.kungahuset.se › royalcourt › royalfamily › hmkin...
HM King Carl XVI Gustaf - Sveriges Kungahus
Carl Gustaf Folke Hubertus, **King of Sweden**, Duke of Jämtland, became Sweden's Head of State on 15 September 1973 at the age of 27, succeeding his ...

www.kungahuset.se › monarchytheroyalcourt › themon...
The Monarchy in Sweden - Sveriges Kungahus
Sweden's King is Carl XVI Gustaf. The Swedish monarchy rests on traditions dating back more than a thousand years. We do not know who the first **kings of** ...

www.britannica.com › biography › Carl-XVI-Gustaf
Carl XVI Gustaf | king of Sweden | Britannica
Apr 26, 2020 - Carl XVI Gustaf, in full Carl Gustaf Folke Hubertus, (born April 30, 1946, Stockholm, Sweden), **king of Sweden** from 1973. Carl XVI Gustaf. Quick ...

www.bbc.com › news › world-europe-49958085
Swedish King Carl Gustaf removes grandchildren from royal ...
Oct 7, 2019 - It noted that **King** Carl XVI Gustaf and his wife Queen Silvia, and Crown Princess Victoria and her husband Prince Daniel were "**Sweden's** highest ...

www.gettyimages.com › photos › carl-xvi-gustaf-king-...
Carl Xvi Gustaf King Of Sweden Stock Pictures, Royalty-free ...

Carl XVI Gustaf of Sweden (Carl XVI Gustaf)
King of Sweden

kungahuset.se/royalcourt/royalfamily/hmkingcarlxvigustaf.4....

Carl XVI Gustaf is the reigning King of Sweden. He ascended the throne on the death of his grandfather, King Gustaf VI Adolf, on 15 September 1973. He is the youngest child and only son of Prince Gustaf Adolf, Duke of Västerbotten, and Princess Sibylla of Saxe-Coburg and Gotha. Wikipedia

Born: April 30, 1946 (age 74 years), Haga Palace, Solna
Full name: Carl Gustaf Folke Hubertus
Children: Victoria, Crown Princess of Sweden, MORE
Siblings: Princess Birgitta of Sweden, MORE
Grandchildren: Princess Estelle, Duchess of Östergötland, MORE
Parents: Prince Gustaf Adolf, Duke of Västerbotten, Princess Sibylla of Saxe-Coburg and Gotha

People also search for · View 15+ more

Queen Silvia of Sweden — Spouse
Victoria, Crown Princess... — Daughter
Prince Carl Philip, D... — Son
Princess Madeleine, Duchess... — Daughter
Prince Daniel, Duke of... — Son-in-law

Figure 2.3.2: Example of Google KG in Search Engine (Source: Google)

technical background related to constructing a KG.

2.3.1 Information Extraction IE

The first step in constructing a company domain KG is to extract information from the 10-K filings using an area of study in NLP called IE. The goal of IE is the extraction of structured information of entities, relationships, and events from unstructured data. The extracted entities can be a person, organization, location, time, and the extracted relations and event can be a competitor, acquisition event, and more [5][6].

Common components for IE consists of general linguistic analysis, which includes Tokenization, Part of Speech (POS) tagging, and Dependency parsing, and domain-specific NER and RE. They are described below.

Sentence Boundary Detection and Tokenization

Sentence Boundary Detection is the process of breaking a document down into sentences by detecting the boundary of each sentence. Tokenization is the process of breaking sentences down into single units, which are called tokens. A token can be a word, a number, or a symbol.

Sentence Boundary Detection and Tokenization in English can be achieved by a rule-based method such as detecting spaces and line breaks between words and defining a continuous string of Latin letters or numbers as a token and detecting period symbol ”.”. However, the challenges in Sentence Boundary Detection is that the period symbol ”.” is used in number such as 0.1 and abbreviation like J.F.K and Fig. [22].

As a result, a statistical approach or hybrid approach can provide higher performance than just a rule-based method [22].

Part-of-speech (POS) tagging

POS tagging is the process of tagging each word in a sentence with its POS. Based on the oxford dictionary, POS is ”a category to which a word is assigned following its syntactic functions.” Common POS in the English language are noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection, and numeral [5].

POS tagging in English has challenges in correctly tagging the right POS to an ambiguous word that can have multiple POS. For example, the word ”set” can be a noun, verb, and adjective.

POS tagging in English can be developed using a rule-based method such as the Brill tagger or a statistical-based method such as Hidden Markov Models to achieve high performance.

Dependency parsing

A sentence can be seen as a composition of phrases, and dependency parsing recognizes the structure of the sentence composition and connects each word with its depending word. The dependency’s common label is subject, object, root, compound, determiner, auxiliary, and adjectival and adverbial modifier [23]. Dependency parsing is a crucial component for developing a more advanced component for IE.

For example, the dependency structure of the sentence from previously mentioned Microsoft SEC 10-K filing, ”On October 25, 2018, Microsoft acquired GitHub, Inc. in a $7.5 billion stock transaction.” can be seen in Figure 2.3.3. The ROOT of the sentence is the verb ”acquired” and the subject to the ROOT is ”Microsoft”. The object is ”Github”. Dependency parsing shows the structure relations between words in the sentence.

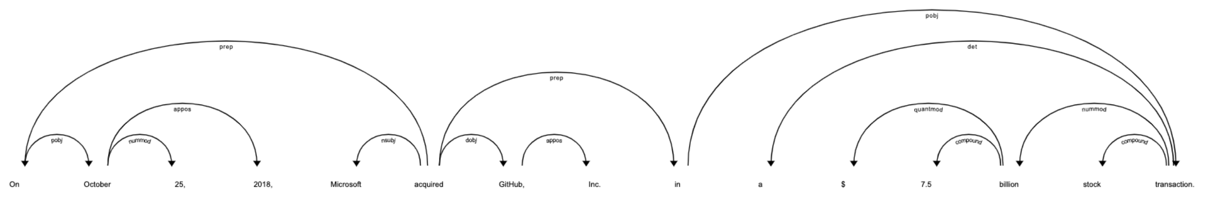


Figure 2.3.3: Example of Dependency parsing (Source: Author using spaCy visualizer)

Named Entity Recognition (NER)

NER is a natural language processing technique that identifies and extracts entities such as a person, organization, location, and time from text [6]. Because entities are nouns, it takes the form either in a single noun or a phrase composed of a list of words. For example, ”Microsoft” and ”GitHub Inc” from the sentence used in 2.3.1 can be considered as Organization type entities.

There are two tasks in NER. First, NER needs to identify entities. Second, second it needs to classify such entity based on the pre-defined entity type. NER can be developed by both rule-based and statistical-based methods.

With the rule-based method, the dictionary/word library can identify a known entity name. Also, linguistic patterns based on POS, dependency parsing, and lexicons can be used to identify an unknown entity. For example, person and organization entities such as Bill Gates and Microsoft Corporation usually contain proper nouns with the first letter capitalized. To classify the identified entity, the rule-based method combined linguistic pattern and regular expression. For example, when ”Inc.” is in a sentence and its previous word’s first letter is capitalized, the capitalized word along with ”Inc.” is likely an entity with the category of Organization [24] [5]. By defining a list of rules, NER can be achieved. However, it can be time-consuming.

With the rise of Wikipedia, where every entity is categorized and updated, a statistical-based method can replace a manually curated list of rules [25]. Notably statistical-based approach for NER is CRF. It can achieve higher performance compared to other supervised machine learning techniques mentioned in Section 2.2. This is evident as many current state-of-the-art NER use CRF [10].

Relation Extraction (RE)

RE is the process of identifying and classifying the relations between two or more entities extracted from documents or sentences.

For example, from the sentence in Section 2.3.1, ”On October 25, 2018, Microsoft acquired GitHub, Inc. in a $7.5 billion stock transaction.”, exists an ”acquired” relationship between the two entity ”Microsoft” and ”GitHub Inc”. RE is a crucial step in forming triples because NER is only able to extract the entities which can only constitute the head and the tail of a triple. Therefore, RE is needed to extract the events or relationships (r) where the entities are mentioned [26].

Generally, the relation extracted from documents or sentences needs to be pre-defined by an ontology to have a good permanence. Current RE focuses on extracting binary relations such as ”acquired” from the example mentioned above. Besides, the more fine-grained relation one wants to extract from the text, the more difficult it is for RE to perform well. Current state-of-the-art-systems perform below the 0.70 F1 measure in extracting event relationships with even worse performance when extracting domain-specific relation [10].

RE can be built by either a rule-based or statistical-based method. With the rule-based method, a list of rules based on keywords, POS tagging, and dependency parsing is used to identify the relationship between two classified entities [26].

Open Information Extraction (OIE)

Because of the limitation of RE extracting pre-defined relation between two entities, OIE method was proposed to allow more general relation extraction. OIE is a process of extracting both entities and their relations from text and formulate the extracted information into tuples at the same time. The first OIE software is TextRunner, which can extract a text document without any human input by using unsupervised learning combined with self-supervised learning [27]. Because OIE does not require a pre-defined rule set and labeling training data by a human, it can provide scalability in constructing a knowledge graph. However, the current state-of-the-art OIE model can only identify relationships from a verb, and the precision of the extracted tuples are not as high as the method combining RE and NER [28].

2.3.2 Knowledge Representation

After IE, the project needs to store the extracted information. This process is related to knowledge representation. Knowledge representation is an area of artificial intelligence (AI) study that focuses on how to store knowledge systematically so that an intelligent machine can ingest the knowledge and infer new conclusions [29] [30].

There are a few ways of representing human knowledge so that intelligent machines can comprehend it and generate new knowledge. The conventional method is called the knowledge base (KB), where one builds a system where each factual knowledge is interlinked with one another. Alternatively, KG is a method to represent knowledge structurally. After entities and corresponding relations are classified using IE described in Figure 2.3.1, triples can be formed. A KG can be viewed as a set of triples, where the nodes are entities, and the edges are relations between entities. Each triple represents an assertion or a fact [7].

Graph database

To construct a Knowledge Graph (KG), a graph database is used. A graph database also allows for the storage, update, deletion, and retrieval of the triples within a KG as time passes by. A graph database stores entities as the nodes, and the edges represent the relations between the nodes. Due to a graph model structure, a graph database can handle complex relationships in connected entities dynamically. [8].

With the graph database, a machine can query the database and utilize the KG in an advanced application. For example, to allow relevant movies and actors to be presented when one searches Tom Hanks, a machine can access a KG graph database to finds relevant entities and relations from the database using graph theory. The extracted entities and relations are from different web sources. The example of a KG can be seen in Figure 2.3.4

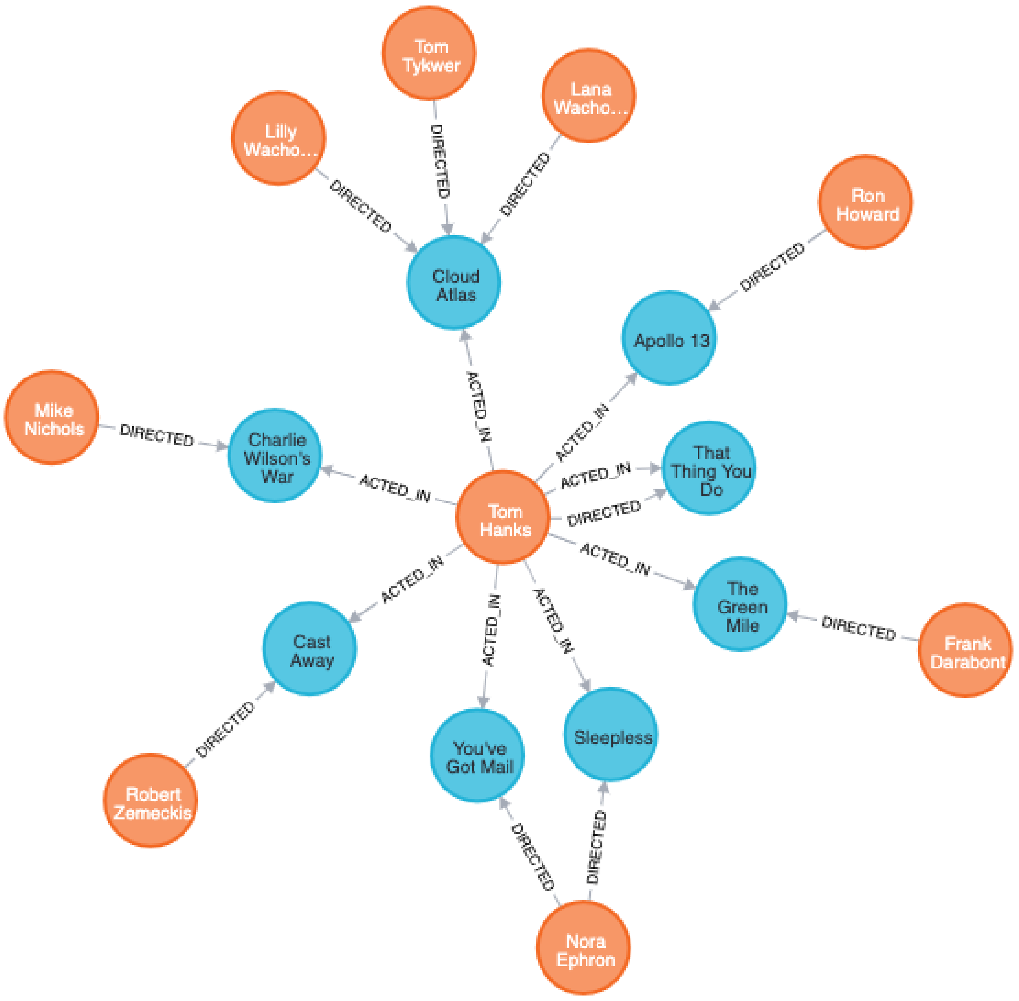


Figure 2.3.4: A sample movie KG from sample data in Neo4j (Source: Neo4j)

2.4 Related Work

In past 10 years, with the introduction of online encyclopedia, such as Wikipedia, there is an increasing number of textual information available. This leads to increasing research development on IE and KG construction to represent knowledge [6] [7] [31].

2.4.1 Named Entity Recognition (NER)

Based on Kejriwal's work, NER is one of the most crucial components in the process of constructing a Domain-Specific KG [10]. Traditionally, NER were developed using a set of rules. With a set of handcrafted rules, entities can be identified and classified by identifying patterns in a sentence that match the rules. However, since the increasing availability of data and computation power, handcrafted rules are replaced by automatic rules determined using supervised machine learning. However, when there is a lack of data to train a statistical model, handcrafted rules are still the go-to method [32].

Besides the rule-based method, NER built using statistical-based methods is the primary research area, and several proposed methods using supervised, semi-supervised machine learning techniques have achieved high accuracy.

Supervised learning is the current state-of-the-art technique for NER. Supervised learning techniques include Hidden Markov Models, Support Vector Machines (SVMs), and Conditional Random Fields (CRFs), where CRF provides higher performance compares to other supervised learning techniques. A large amount of annotated training data is needed to build a supervised learning model where the model can identify entities provided in the training data and create disambiguation rules based on discriminative features [10] [32].

Although NER using supervised learning techniques mentioned in [32][10] can achieve high accuracy, there is a large amount of training data labeled for general purpose. The current advancement of general-purpose NER based on the statistical method mentioned above is because there is a large amount of exiting annotated data. However, there is a limited amount of annotated data in a domain-specific area such as medicine, chemistry, academia. As a result, currently, publicly available pre-trained NER cannot identify and extract those entities. Thus, to develop a high-performance domain-specific NER using supervised learning, one would still require to obtain a large amount of training data and spend time on annotating the training data.

However, semi-supervised learning techniques can reduce human annotation time on training data significantly. Semi-supervised learning techniques such as bootstrapping and mutual bootstrapping can be used to build NER[33]. It can reduce the time by initially having a simple model based on a smaller set of annotated data or seeds in the learning process. As the model bootstraps through the learning process, it adds to the newfound entities. Although semi-supervised learning can reduce human time spent on annotating training data, the supervised learning method still outperforms semi-supervised learning methods.

Because of the reduced amount of annotated training data required to build NER using semi-supervised learning, Schon [34] proposed to use the bootstrapping technique to obtain domain-specific business products by defining a set of lexical patterns to identify the mention of Company Provides Product as the initial seeds. However, Schon did not disclose the evaluation of his result.

Besides using either supervised or semi-supervised learning techniques to train a domain-specific NER model, some researchers focus on the hybrid method of a statistically based method combining with a rule-based method. Mikheev, Grover, and Moens's work used the hybrid approach to identify person, organization, location entities in the general context, which have an F1-score of 93.4% [35]. Rocktäschel [36] combined CRF technique with a rule-based dictionary to create a hybrid method that achieves an F1-score of 68.1%, which outperforms an open-source chemical NER tool, OSCAR4, by 10.8 percentage points. The training and testing data used for the CRF are from a pre-annotated corpus.

Because popular open-source pre-trained NER can recognize entities such as Location, Person, and Organization, Stewart, Enkhsaikhan, and Liu's work [37] combined pre-trained open-source models with rule-based chunking to extracts car company domain specific KG. Their work extracts not only car company, car model entities but also salient information units such as ”an American multinational automaker” and ”a suburb of Detroit”. However, the paper focuses on using rule-based noun chunking to identify salient information such as ”a suburb of Detroit” and ”an American multinational automaker” but was not used to improve the NER performance in identifying real entities. Due to the lack of pre-tagged data in automotive domain, Rubens and Agarwal's work combined open-source NER with and rule-based dictionaries to build a handcrafted automotive domain NER to extract automotive information which yields to F1-score of more than 90% [38].

Furthermore, Siencnik's work [39] explored using word embedding or word vectors method to find similarity between words to improve an existing NER performance with a small amount of data annotation.

2.4.2 Summary

This project will not focus on developing a KG using supervised learning NER as it requires a large amount of domain-specific annotated data that are not publicly available. Also, the significant amount of time spends on human annotation defeats the project's purpose of decreasing human input. Furthermore, the project does not use semi-supervised learning NER since its performance currently cannot beat the supervised learning model.

Schon's proposed method of using a statistical-based CRF method combined with a rule-based dictionary method shows the appealing result compared to just a statistical-based method [36]. Stewart, Enkhsaikhan, and Liu's work [37] that combine pre-trained open-source model and rule-based method lack of analysis of the performance. Siencnik's work shows that it is possible to use word embedding for entity classification.

To summarize, this project we have chosen to explore using hybrid methods of pre-trained open-source NER with a set of rule-based methods to extract entities and build KG. The reason for using pre-trained NER with a set of the rule-based method is because:

1. There is a lack of company and product domain-specific corpora for training statistical models.
2. It could meet the goal of reducing the human input time by eliminating the human annotation time.
3. It could be a solution to the second problem of the project listed in Section 1.2.
4. A purpose of this project is to check the hybrid performance since there is a lack of research on the analysis of the hybrid approach performance in building company domain-specific KG.

Chapter 3

Methodologies

This chapter describes the framework used to perform the research and analyse which tools should be used in a different stage of the framework. First, the framework is introduced in Section 3.1. Second, the data format and source are described along with the tools that were used to collect them in Section 3.2. Next, a step by step procedure is given on how to pre-process the data, how to extract entities and relations from the data, how to visualize and store the extracted information as a KG, and how to evaluate the performance in the order of Section 3.3, 3.4, 3.5. Lastly, the validity and reliability of the data are in Section 3.6.

3.1 Framework

This project used the CRoss Industry Standard Process for Data Mining (CRISP-DM) framework to construct KG. CRISP-DM framework consists of the following steps and is shown in Figure 3.1.1 [40]:

1. **Business Understanding** stage focuses on understanding the problems and objects, and defining the technical goals of the project based on the problems.
2. **Data Understanding** stage focuses on understanding the data source, how to collect data from the data source, analyzing the data quality, and get insights from the acquired data.
3. **Data Preparation** stage focuses on cleaning and pre-processing the data so that it can be ready for the modeling.

4. **Model Building** stage focuses on applying modeling techniques and adjusting parameters to obtain accurate results.

5. **Model Evaluation** stage focuses on evaluating and analyzing the quality of the model so that it meets the objects defined in the Business Understanding stage.

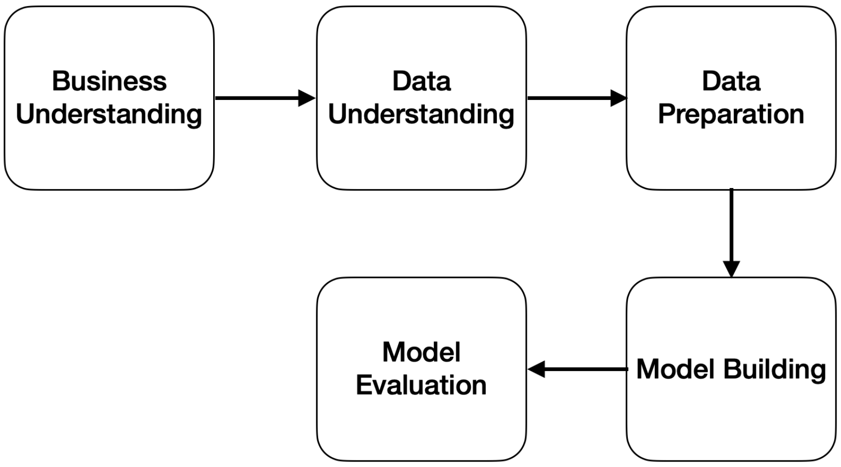


Figure 3.1.1: Framework (Source: Author, adapted from [40])

3.2 Data Understanding

This section describes the project's data source, where the data are stored, and evaluate which HTML Web crawler is best suited to acquire the data.

Data source

The first step in constructing a generic or domain-specific KG is to acquire raw data that contains the relevant knowledge of the topic. The data should be corpus and is usually from the following sources: Webpages, database, offline text, and the document types can be the text, words, CSV, Excel, etc.

In this project, the data is SEC 10-K filings, and SEC filings are stored in Electronic Data Gathering, Analysis, and Retrieval system (EDGAR). SEC filings can be publicly accessed through the SEC's HTTPS file system. To search a SEC filings that belong to a specific company, one would need to input either the company's Central Index Key (CIK) or ticker/stock symbol [16].

HTML Web crawler

Since SEC filings can only be accessed publicly through HTTP protocol, the data source is accessed through a web browser with HTML tags used to render the content. Therefore, it requires an HTML Web crawler to access and download the data.

There are a few existing tools and libraries for downloading the SEC fillings:

1. General-purpose web crawler: Nutch and Scrapy can be used to crawl the different webpage at scale and select a certain part of the HTML based on the HTML tag. For example, extracting table contents based on <table> tag. The advantage of a General-purpose web crawler such as Nutch and Scrapy is that it can acquire any web data with an URL or a domain of URLs.

2. Tailored SEC web crawler: sec-edgar-downloader, python-edgar, and edgar are three of the licensed open source python libraries tailored to crawl SEC's EDGAR systems. Instead of having an URL as an input, the three libraries allow the user to acquire the fillings by taking a company's Central Index Key (CIK) or ticker/stock symbol. However, the libraries' downside is that one cannot extract a specific part of the HTML based on the HTML tags. This means that tailored SEC web crawlers extract the whole content of a specific HTML page [41].

Due to the project's investigative nature (i.e not focused on scalability) and since the data source is only from the SEC's EDGAR system, this project chose to use a python public library sec-edgar-downloader to acquire the required SEC Edgar Filling.

3.3 Data Preparation

This section describes how the data is prepared so that the it can be used in the data modeling stage in Section 3.4. In the data preparation stage, all the unwanted content such as HTML tags, page number, heading, etc. are removed using HTML parser and regular expression.

3.3.1 HTML Parser

A HTML parser is a process of removing HTML tags and transforming HTML documents into readable text data. Below are three of the existing Python HTML parser libraries:

1. BeautifulSoup is an open-source library designed to efficiently modify, navigate, search, and pull data from HTML, XHTML, and XML files.

2. html.parse is the basis module to extract data from HTML, XHTML files.

3. lxml is the basis module to extract data from HTML and XML files.

One needs to observe HTML's DOM Tree, tag, tag's attributes, and tag's attribute's value to remove unwanted content. For example, to divide filing into different sections, the project identifies an anchor tag with an attribute value of Hypertext Reference from the table of content (<a href="Value"). After the href attribute value is obtained, the project tries to find the corresponding section by identifying the anchor tag with the name attribute that has the same value.

Because of the need to navigate, search, and extract specific data from the data, this project chose to use BeautifulSoup.

3.3.2 Regular Expression

Besides HTML parser, regular expression is another tool to clean and remove the certain data so that the cleaned data can be ready for data modeling. Regular expression is a rule-based method that filters desired data by matching specific patterns from the data. This can be achieved by using the Python re library.

After observing the data obtained using HTML parsing, the data contains nonvalue adding content that may increase the NLP processing time. As a result, this project chose to use regular expression to remove unwanted data to improve the processing time in the data modeling stage.

3.4 Model Building

In this project, the objectives of data modeling are to extract entities and relations between each entity from the data. These are achieved by using IE's NER and RE components.

3.4.1 NER

As mentioned in Section 2.3.1, NER model can be built using rule-based methods. However, building a rule-based model is a time-consuming process that requires one to thoroughly analyze the sentence structure, word dependency to extract entities. This requires one to be an expert in linguistics. NER model can be built using statistical-based NER methods, which require a large amount of labeled training data and the model training time can be longer depending on the computing power.

This project chose to use an existing pre-trained statistical model for two reasons:

1. Using existing pre-trained statistical model eliminates the time for manual labeling data tasks, and training the model
2. There is an existing pre-trained open-source statistical model that can achieve accuracy close to the state-of-the-art model.

However, existing pre-trained models are designed for general content. Thus, the existing statistical pre-trained model's accuracy is not optimal within this project's domain data. Therefore, rule-based patterns using POS tagging and dependency parsing are also used to identify and classify the entities which the pre-trained model cannot identify [42].

Below is the comparison of three popular open-source NLP tools [43]:

Stanford CoreNLP

Stanford CoreNLP is an open-source Java package developed by The Natural Language Processing Group at Stanford University. Stanford CoreNLP contains NER model that is based on linear-chain Conditional Random Field. Stanford CoreNLP can extract entities class of Location, Person, Organization with the overall F1 score of 70.8% [44]. Besides classifying Location, Person, and Organization, Stanford CoreNLP has other two NER models [44]:

- 3 class model can recognize ”Locations, Persons, Organizations, and Miscellaneous entities”
- 7 class model can recognize ”Locations, Persons, Organizations, Times, Money, Percents, and Dates entities”

spaCy

spaCy is an open-source Python package developed by Matthew Honnibal and Ines Montani from Explosion, a software company. spaCy's NER model is based on a convolutional neural network. The spaCy model can extract entities class of Location, Person, Organization, Product, and more. spaCy NER model has the overall F1 score of 57.9% [43]. Additionally, spaCy allows one to train one's model with labeled data and includes a generic matcher to allow greater flexibility.

NLTK

NLTK is an open-source Python package that is popular in the research community. NLTK NER is based on uses Maximum Entropy Classifier, a supervised machine learning algorithm. NLTK's NER classifier can extract entities class of Location, Person, Organization, and more. NLTK's NER model has an overall F1 score of 0.4734 [43].

Based on the F1 score between the three open-source tools, it is clear that Stanford CoreNLP performs the best in identifying and classifying Location, Person, Organization. However, since this project aims to extract infomration related to companies and their product, spaCy has been chosen for NER. Also, spaCy provides additional flexibility to incorporate rule-based pattern to modify the pre-trained NER model [45].

3.4.2 RE

The relations this project wishes to extract are specific relations at the sentence level. Therefore, the project's focus will be on building a rule-based model that uses regular expression and pattern matching by using POS, dependency parsing, and the improved NER components to analyze sentence structure and extract specific relations between classified entities in a sentence.

Based on the benchmark results from [45], spaCy's dependency parsing's F1 score is 0.9448, and the POS's F1 score is 0.972, and dependency parsing is the fastest compares to Stanford CoreNLP and NLTK. As a result, this project chose to use spaCy's dependency parsing and POS tagging for RE.

3.5 KG formation

As mentioned in Section 2.3, to represent the extracted triples, a graph database is required. The followings are two of the popular open-source graph database:

Neo4j

Neo4j is an open-source, NoSQL, native graph database. The graph data is queried by Cypher, a declarative query language similar to SQL, but simpler [46]. It provides a Python library for graph operations and a native browser to visualize the data stored within [8].

Cayley

Cayley is developed by Google, which provides high compatibility with different data formats, supports different databases and query languages, and allows customization as it is modular [47].

After reviewing the above graph database, there is no significant difference between them that could affect the project's outcome. However, since Neo4j provides a native visualization browser, this project chose to use Neo4j.

3.6 Model Evaluation

Since the project can be seen as a classification problem on whether the extracted triples are correct or not, the proposed solution's performance is evaluated using the Precision, the Recall, and, the F1-score metrics described below. Figure 3.6.1 visualizes the Recall and Precision for a better understanding of the calculation:

- Precision: Precision is the ratio between correct triple findings and the total triple found findings. It is the measure of the correctness of the solution.

$$Precision = \frac{CorrectFindings}{TotalFindings}$$

- Recall: Recall is the ratio between correct triple findings and the expected findings. It is the measure of how completeness of the solution is.

$$Recall = \frac{CorrectFindings}{ExpectedFindings}$$

- F measure: F measure is the weight average of the Recall and Precision and it is common to set β= 1 to give Recall and Precision equal weight to calculate the F1 measure [48].

$$F = \frac{(\beta^2 + 1) * Precision * Recall}{(\beta^2 * Precision) + Recall}$$

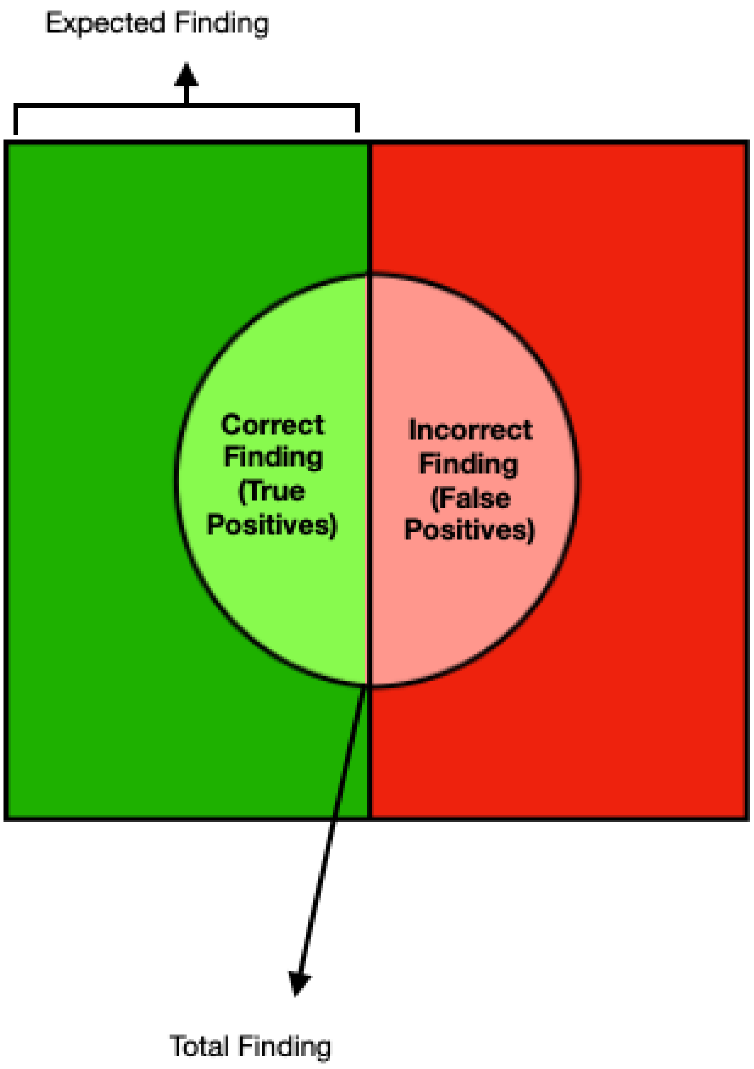

Expected Finding
Correct Finding (True Positives)
Incorrect Finding (False Positives)
Total Finding

Figure 3.6.1: Recall and Precision (Source: Author, adapted from [49])

3.7 Validity and Reliability of The Data

Multiple acquisitions on the same data were performed to ensure reliability in data acquisition. The same data was also run multiple times to ensure the reliability of the data preparation.

The information gathered in the project from the SEC EDGAR system are monitored at every stage of the development to ensure the validity of the data.

Chapter 4

Building a Domain-Specific KG

This chapter describes the design of the domain specific KG using data from SEC 10-K filings and the process of building it. Section 4.1 describes the general structure and information about the data, the SEC 10-K filing. Section 4.2 describes the system architecture and ontology design. Section 4.3 provides the insight about the acquired data. Section 4.4 describes the steps to pre-process and clean the data. Section 4.5 describes the IE model along with example and also describes how the triples are formed. Section 4.6 describes how the triples are stored and presents an example of the KG.

4.1 Data Understanding

The data used in the project is SEC 10-K filing, which is an unstructured text corpus. The filing consists of a total of fifteen items, which are categorized into four parts:

- **Part One** contains information about the company's business, market, competitor, different risk factors that may affect the company business, legal proceedings, and physical assets such as properties.
- **Part Two** contains information about the company's financial data such as auditor's report, management's report on the internal control over the financial reporting, financial statements, and equity holdings.
- **Part Three** contains information about the higher management structure and information about the higher management, such as executive's compensation, security ownership, transaction, and accounting fees.

- **Part Four** contains a summary of the 10-K filing and executives' signature.

Because the project's goal is to build a company domain knowledge graph that contains information about company-related entities and companies' relationships with other companies, which can be found in Part One of the report, Part Two to Four of the report will not be considered.

4.2 System Architecture

The system architecture for constructing a domain-specific KG is shown in Figure 4.2.1.

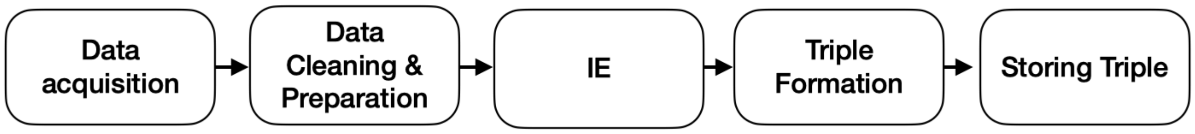


Figure 4.2.1: System Architecture (Source: Author)

The input data to the system is SEC 10-K filings from SEC EDGAR database, and the output is a graph containing extracted triples. The function of each component in the system architecture is described below:

1. **Data Acquisition** downloads the SEC 10-K filings from the data source.
2. **Data Cleaning & Preparation** clean and remove unwanted parts of the filings so that it can be ready for IE.
3. **IE** extracts entities and relations from the cleaned data. This project uses spaCy's en_core_web_lg model to perform IE. The model is under the CC BY-SA 3.0 license.
4. **Triple Formation** builds triples from the extracted entities and relations.
5. **Storing Triples** stores the triples to the graph database Neo4j.

4.2.1 Ontology Design

This project's ontology focuses on extracting two types of entities and three types of relations related to a company. The entity types are: Company type and Product type, and the relationships are has_product, has_competitor, and acquire. The ontology is presented in Figure 4.2.2

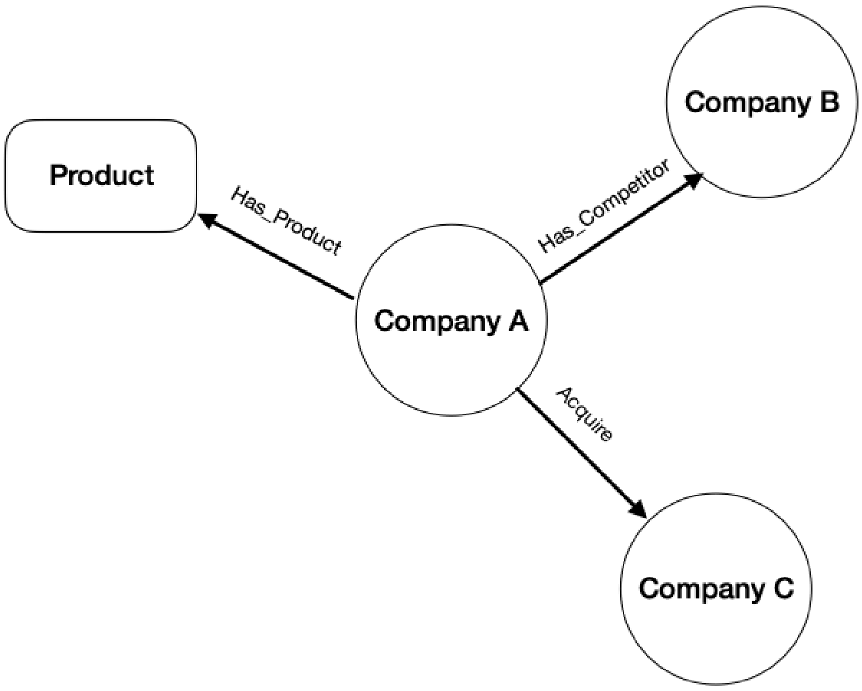
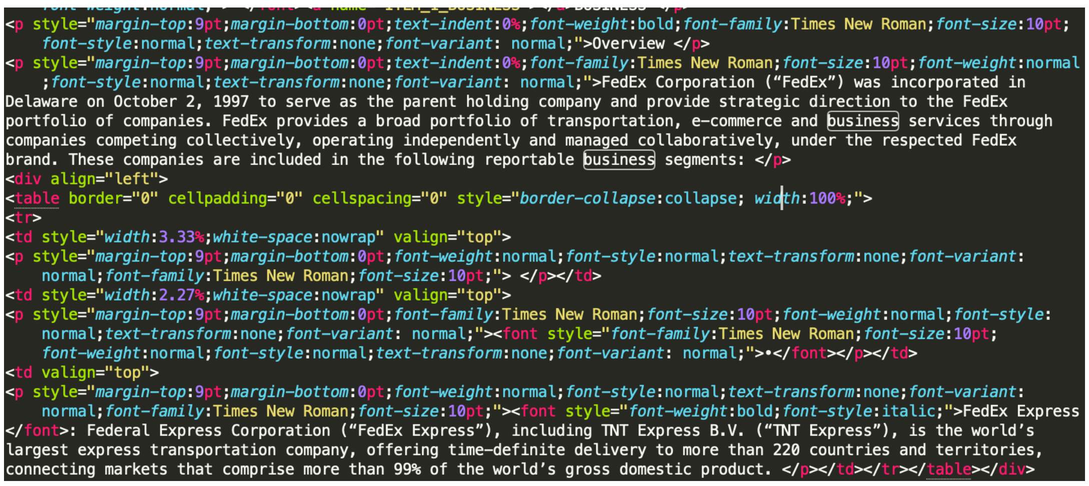


Figure 4.2.2: Company Ontology (Source: Author)

4.3 Data Acquisition

The project downloads 20 SEC latest 10-K filings for 20 different companies from the SEC EDGAR system using the python HTML crawler sec-edgar-downloader library with MIT License (MIT). Ten 10-K filings are used to build the hybrid model, and the remaining ten 10-K filings are used to evaluate the model's performance. A screenshot of a part of a downloaded filing can be seen in Figure 4.3.1.



```
<p style="margin-top:9pt;margin-bottom:0pt;text-indent:0%;font-weight:bold;font-family:Times New Roman;font-size:10pt;
    font-style:normal;text-transform:none;font-variant: normal;">Overview </p>
<p style="margin-top:9pt;margin-bottom:0pt;text-indent:0%;font-family:Times New Roman;font-size:10pt;font-weight:normal
    ;font-style:normal;text-transform:none;font-variant: normal;">FedEx Corporation (“FedEx”) was incorporated in
Delaware on October 2, 1997 to serve as the parent holding company and provide strategic direction to the FedEx
portfolio of companies. FedEx provides a broad portfolio of transportation, e-commerce and business services through
companies competing collectively, operating independently and managed collaboratively, under the respected FedEx
brand. These companies are included in the following reportable business segments: </p>
<div align="left">
<table border="0" cellpadding="0" cellspacing="0" style="border-collapse:collapse; width:100%;">
<tr>
<td style="width:3.33%;white-space:nowrap" valign="top">
<p style="margin-top:9pt;margin-bottom:0pt;font-weight:normal;font-style:normal;text-transform:none;font-variant:
    normal;font-family:Times New Roman;font-size:10pt;"> </p></td>
<td style="width:2.27%;white-space:nowrap" valign="top">
<p style="margin-top:9pt;margin-bottom:0pt;font-family:Times New Roman;font-size:10pt;font-weight:normal;font-style:
    normal;text-transform:none;font-variant: normal;"><font style="font-family:Times New Roman;font-size:10pt;
    font-weight:normal;font-style:normal;text-transform:none;font-variant: normal;">•</font></p></td>
<td valign="top">
<p style="margin-top:9pt;margin-bottom:0pt;font-weight:normal;font-style:normal;text-transform:none;font-variant:
    normal;font-family:Times New Roman;font-size:10pt;"><font style="font-weight:bold;font-style:italic;">FedEx Express
</font>: Federal Express Corporation (“FedEx Express”), including TNT Express B.V. (“TNT Express”), is the world’s
largest express transportation company, offering time-definite delivery to more than 220 countries and territories,
connecting markets that comprise more than 99% of the world’s gross domestic product. </p></td></tr></table></div>
```

Figure 4.3.1: Example of a 10-K raw file (Source: Author)

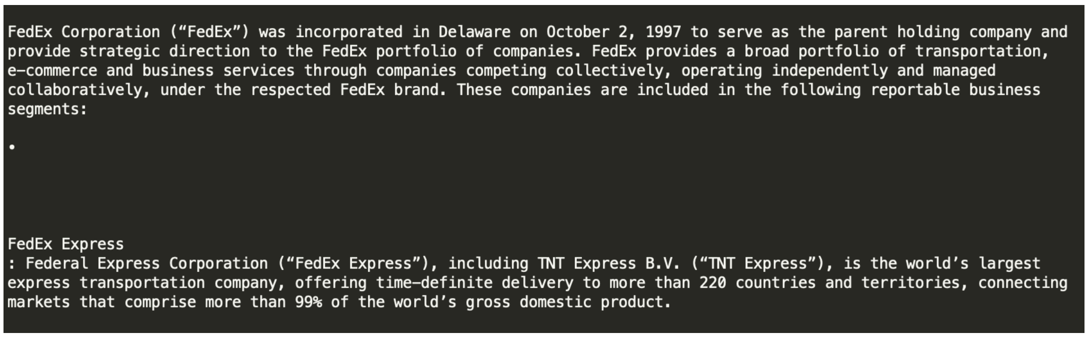
4.4 Data Cleaning and Preparation

Because the project acquired data using a tailored SEC web crawler, the data contain HTML tags with CSS attributes that need to be removed. After inspecting the acquired data, the sec-edgar-downloader acquired the 10-K filings and their related exhibits and graphs in the XML. Therefore, the unwanted part of the fillings, such as attachment, XML, Excel, appendix, etc. needs to be removed after the data are acquired. Also, because the project's focus on extracting the company's relationship in terms of products, subsidiary, and competitor can be found in Part One of the filing, Part Two to Four of the filling are removed. The contents are removed using BeautifulSoup.

Finally, there are some minor formatting issues after the above contents are removed, which can affect the IE process. Therefore, formatting was performed using regular expression.

4.4.1 HTML parser

Since 10-K filings contain a table of content that uses <a> tag with reference attribute to reference different parts of the filing, the project extracts Part One of the filing based on the reference id using BeautifulSoup. A sample result of the data cleaning and preparation using BeautifulSoup is provided in Figure 4.4.1. When comparing the raw 10-K filing in Figure 4.3.1 with the cleaned 10-K filing in Figure 4.4.1, one can observe all the HTML tags and CSS attributes are removed, and thus the only text is presented.


```
FedEx Corporation (“FedEx”) was incorporated in Delaware on October 2, 1997 to serve as the parent holding company and
provide strategic direction to the FedEx portfolio of companies. FedEx provides a broad portfolio of transportation,
e-commerce and business services through companies competing collectively, operating independently and managed
collaboratively, under the respected FedEx brand. These companies are included in the following reportable business
segments:

•



FedEx Express
: Federal Express Corporation (“FedEx Express”), including TNT Express B.V. (“TNT Express”), is the world’s largest
express transportation company, offering time-definite delivery to more than 220 countries and territories, connecting
markets that comprise more than 99% of the world’s gross domestic product.
```

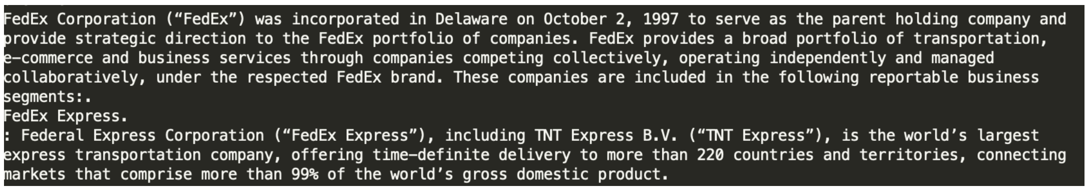
Figure 4.4.1: Example of a 10-K clean file (Source: Author)

4.4.2 Regular Expression

One can also see that in Figure 4.4.1, the raw text contains unwanted contents. The project observed that the unwanted contents could be page number, multiple newlines, spaces, tabs, and non-alphanumeric values such as bullet points. All of these contents can affect the IE process and thus need to be removed. Additionally, some of the sentences are split into two lines with a line breaker in some cases due to the structure of the original HTML data. This may lead to sentence boundary detection not working correctly in the IE stage and can lead to incorrect information extraction or inability to extract information. Therefore, in addition to using BeautifulSoup to remove HTML tags and CSS attributes, the project also uses regular expression python library re to remove the unwanted contents.

Furthermore, due to the nature of a legal document, the filings sometimes refer to the company name using ”We”, ”The company”, or ”our”. This also affects the IE to extract the company entity from the filing. The project uses a regular expression to replace the ”We”, ”The company”, or ”our”, with the company name.

A sample result of the preparation using regular expression is provided in 4.4.2.



```
FedEx Corporation (“FedEx”) was incorporated in Delaware on October 2, 1997 to serve as the parent holding company and
provide strategic direction to the FedEx portfolio of companies. FedEx provides a broad portfolio of transportation,
e-commerce and business services through companies competing collectively, operating independently and managed
collaboratively, under the respected FedEx brand. These companies are included in the following reportable business
segments:.
FedEx Express.
: Federal Express Corporation (“FedEx Express”), including TNT Express B.V. (“TNT Express”), is the world’s largest
express transportation company, offering time-definite delivery to more than 220 countries and territories, connecting
markets that comprise more than 99% of the world’s gross domestic product.
```

Figure 4.4.2: Example of a finished cleaning file (Source: Author)

4.5 IE

IE is used here to extract the relations and entities from the filings. This project’s IE pipeline consists of tokenization, POS Tagging, Dependency Parsing, NER, and RE as shown in Figure 4.5.1

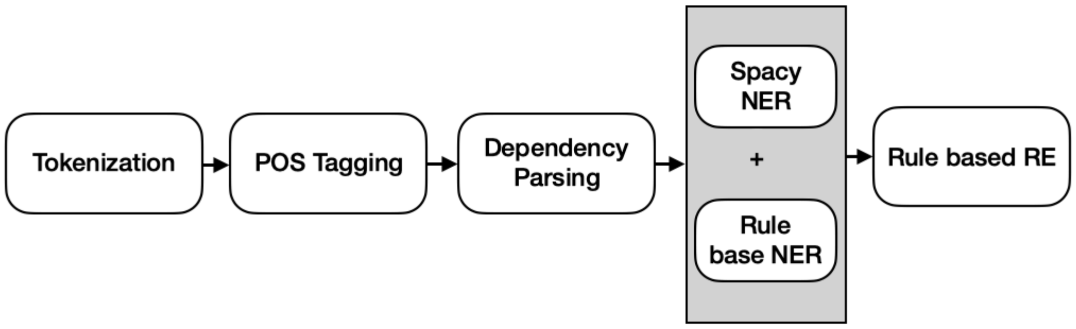


Figure 4.5.1: IE pipeline (Source: Author)

To illustrate the process of each component in the IE, this section uses the following sentences from Microsoft and Amazon's 10-K filing:

- Sample sentence 1: "Amazon.com seeks to be Earth's most customer-centric company."
- Sample sentence 2: "Dynamics competes with vendors such as Infor, NetSuite, Oracle, Salesforce.com, SAP, and The Sage Group to provide cloud-based and on-premise business solutions for small, medium, and large organizations."
- Sample sentence 3: "Microsoft 365 brings together Office 365, Windows 10, and Enterprise Mobility + Security to help organizations empower their employees with AI-backed tools that unlock creativity, increase teamwork, and fuel innovation, all the while enabling compliance coverage and data protection."
- Sample sentence 4: "On November 16, 2018, LinkedIn acquired Glint, an employee engagement platform, to expand its Talent Solutions offerings."
- Sample sentence 5: "Microsoft's system management solutions compete with server management and server virtualization platform providers, such as BMC, CA Technologies, Hewlett-Packard, IBM, and VMware."
- Sample sentence 6: "Microsoft's Search business, including Bing and Microsoft Advertising, is designed to deliver relevant online advertising to a global audience."
- Sample sentence 7: "Microsoft's Enterprise Mobility + Security offerings also compete with products from a range of competitors, including identity vendors, security solution vendors, and numerous other security point solution vendors."

4.5.1 Tokenization

Tokenization segments sentences from documents and words from sentences. Thus, tokenization is the fundamental part of any NLP, including IE. It is needed so that other components in the pipeline can work properly.

This project uses the spaCy tokenizer to tokenize every sentence and words. spaCy tokenizer has token attributes such as pos_, dep_, ent_type_, which integrates well with other IE components. spaCy token's attributes allow spaCy to automatically assign each of the corresponding attributes after POS tagging, Dependency parsing, and spaCy NER are performed.

The TEXT and LEMMA columns in Table 4.5.1 presents the output of word tokenization using Sample sentence 1.

4.5.2 POS tagging

Once the tokenizer tokenizes words from the documents, POS Tagging is performed. As mentioned in Section 2.3.1, POS tagging categorizes every word to its POS. POS tagging is used in the IE pipeline because it is crucial to identify noun and verb which is used to identify entities and relations when doing rule-based NER and RE.

spaCy's POS Tagging comes with two labels: POS and TAG. POS label is based on Universal POS tags from Universal Dependencies, an open community for consistent grammar annotation. TAG is from the OntoNotes5 version of the Penn Treebank tag set [45]. TAG label provides a more detailed part-of-speech tag compare to POS.

The POS TAG columns in Table 4.5.1 is the output of the POS tagging using Sample sentence 1. One can see the text ”to”'s and ”s”'s POS are ”PART” but their TAGs are ”TO” and ”POS”. ”TO” means infinitival “to” and ”POS” means possessive ending.

4.5.3 Dependency Parsing

spaCy dependency parsing is performed to identify the structure of sentence composition and word's relation with other words within it. By knowing words relations with other words, phrases, noun chunks in a sentence can be identified. Noun chunk is a noun with more than one word. Therefore, this project can refine the entities and relations by including other words related to them.

Table 4.5.1: Tokenizer and POS tagging

TEXT	LEMMA	POS	TAG
Amazon.com	Amazon.com	PROPN	NNP
seek	seek	VERB	VBP
to	to	PART	TO
be	be	AUX	VB
Earth	Earth	PROPN	NNP
’s	’s	PART	POS
most	most	ADJ	JJS
customer	customer	NOUN	NN
-	-	PUNCT	HYPH
centric	centric	NOUN	NN
company	company	NOUN	NN
.	.	PUNCT	.

For example, dependency parsing is performed on Sample sentence 1, and one can see the structure relations between the words in Figure 4.5.2.

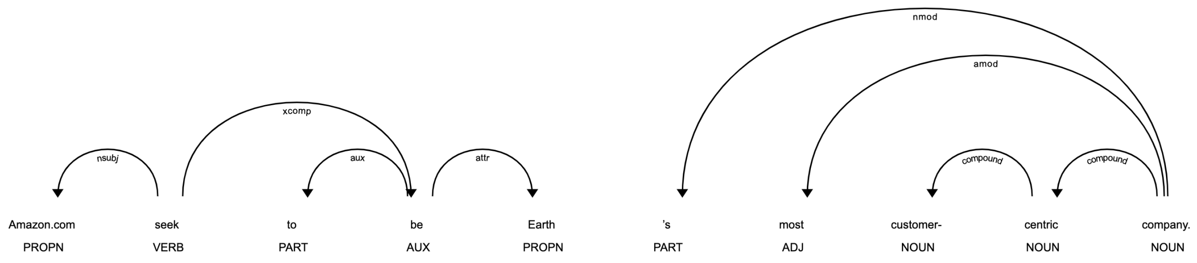


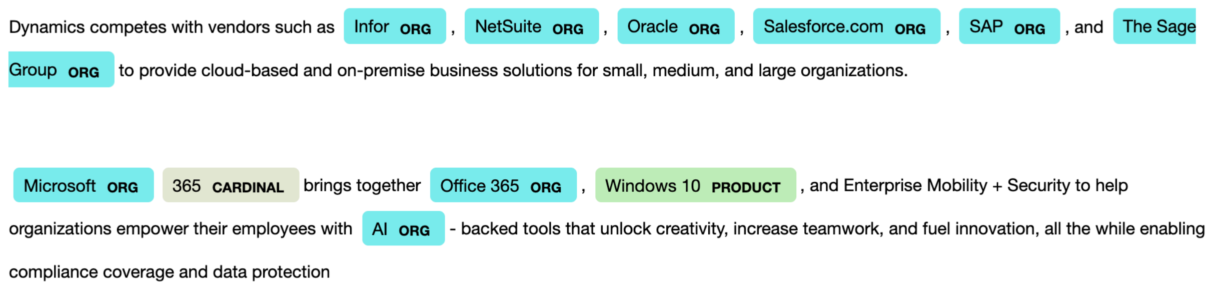
Figure 4.5.2: Dependency Parsing (Source: Author using spaCy visualizer)

4.5.4 NER

This project NER model consists of spaCy NER from en_core_web_lg model and rule-based NER using spaCy Matcher to create lexical patterns using POS tags and dependency parsing labels, _noun_chunks, and word vector to identify and classify new entities.

spaCy NER

spaCy's pre-trained NER model is used to identify entities from the filings. In Figure 4.5.3, spaCy NER is performed on sample sentence 2 and 3 to identify Organization and Product entities. However, one can observe that spaCy NER did not identify all the entities in the sentences. For example, spaCy NER cannot identify Microsoft's products such as Dynamics from the second sentence and Enterprise Mobility + Security in Sample sentence 3. Also, it misclassified Office 365 as an organization and misidentified Microsoft 365 as an organization with a cardinal.



Dynamics competes with vendors such as Infor ORG , NetSuite ORG , Oracle ORG , Salesforce.com ORG , SAP ORG , and The Sage Group ORG to provide cloud-based and on-premise business solutions for small, medium, and large organizations.

Microsoft ORG 365 CARDINAL brings together Office 365 ORG , Windows 10 PRODUCT , and Enterprise Mobility + Security to help organizations empower their employees with AI ORG - backed tools that unlock creativity, increase teamwork, and fuel innovation, all the while enabling compliance coverage and data protection

Figure 4.5.3: spaCy NER (Source: Author using spaCy visualizer)

Rule-based NER

The rule-based method is used to solve the problem done by spaCy as mentioned above.

To identify entities, the project identifies noun chunks that should be entities but were not identified by spaCy NER by first using spaCy's _noun_chunks method to identify all the noun chunks.

After noun chunks are identified, the project needs to filter noun chunks that have been identified by spaCy and noun chunks that have not. After identifying all the noun chunks are entities and are not included in spaCy's entity list, which means they are entities that spaCy cannot identify, the entities need to be classified.

For classification, this project uses the following method mentioned in [42] to classify the identified entities

- Internally pattern, keywords, regular expression are used to classify entity by words within a noun chunk. For example, when there is the mention of "Management", "Platform", "Server", "System", the entity will be classified as

Product and when there is the mention of ”Inc”, ”LLC”, ”Corp”, ”.com”, the entity will be classified as Company.

- External lexicon pattern checks the surrounding of an entity in a sentence. For example, when there is ”’s” before the entity, such as Sample sentence 7, ”Microsoft’s Enterprise Mobility + Security offerings also compete with ...”, then the entity ”Enterprise Mobility + Security” will be classified as Product.
- ”Partial matching” is the third stage of the rule-based NER where all the unclassified entities are compared with the classified entity. For example, Windows 365 is classified as a product because it partially matches Windows, a classified product entity.

In addition to the above methods, word vectors were used to classify the remaining noun chunks by comparing the similarity between identified entities and noun chunks. Observations were made to decide the similarity threshold for entity classification. If a noun chunk’s similarity exceeds the observed threshold, a Product or Organization label classification will be assigned. Since this project’s ontology focus on finding company and product entities, other types of entity similarity were not compared. Algorithm 1 for rule-based NER is presented below:

Once entities are identified and classified, patterns are created, and spaCy’s entity_ruler is used to create rule-based patterns and add into the pipeline.

Algorithm 1 Rule-based NER algorithm

```
1: procedure Name Entity Extraction(document)
2:     Get $NounChunksList$ using spaCy _noun_chunks
3:     Get $EntityList$ using spaCy NER
4:     $RuleBasedEntityList$ =emptyList()
5:
6:     for Each $Chunk$ in $NounChunkList$ do            ▷ Internal & External Pattern Matching
7:         if ($Chunk.root$ == $PROPN$) & ($Chunk$ not in $EntityList$) then
8:             if $Chunk.POS$ match $companyentitypatterns$ then
9:                 $RuleBasedEntityList$.append($Chunk$ with $Companylabel$ )
10:            end if
11:            if $Chunk.POS$ match $productentitypatterns$ then
12:                $RuleBasedEntityList$.append($Chunk$ with $Productlabel$)
13:            end if
14:        end if
15:    end for
16:
17:    for Each $Chunk$ in $RemainingNounChunkList$ do            ▷ Partial Matching
18:        if ($Chunk.root$ == $PROPN$) & ($Chunk$ not in $EntityList$) then
19:            if $Chunk$ partial match $RuleBasedEntityList$ then
20:                $RuleBasedEntityList$.append($Chunk$ with the matched entity label)
21:            end if
22:        end if
23:    end for
24:
25:    for Each $Chunk$ in $RemainingNounChunkList$ do            ▷ Word Vectors
26:        if ($Chunk.root$ == $PROPN$) & ($Chunk$ not in $EntityList$) then
27:            for $entity$ in $EntityList$ or $RuleBasedEntityList$ do
28:                if $entity$ label is $CompanyLabel$ then
29:                    $CompanySimilarity$=getCompanySimilarity($Chunk$,$entity$)
30:                    if $CompanySimilarity$>threshold then
31:                        $RuleBasedEntityList$.append($Chunk$ with $Companylabel$ )
32:                    end if
33:                end if
34:                if $entity$ label is $ProductLabel$ then
35:                    $ProductSimilarity$=getCompanySimilarity($Chunk$,$entity$)
36:                    if $ProductSimilarity$>threshold then
37:                        $RuleBasedEntityList$.append($Chunk$ with $Productlabel$ )
38:                    end if
39:                end if
40:            end for
41:        end if
42:    end for
43:    return $RuleBasedEntityList$
44: end procedure
```

4.5.5 RE

This project's Relation Extraction uses one of the methods from [26] to extract specific relations at a sentence level by finding the occurrence of the relationship between two entities (triples patterns) defined by the ontology. At the granular level, this RE method consists of two tasks. First, RE identifies sentences that contain the predefined relation. After a relationship is identified in a sentence, it checks if the sentence contains an entity pair pattern that matches the triple patterns.

Identify Relation

To extract the three relations defined in the ontology at a sentence level, this project first uses regular expression to identify keywords related to those three relations. The keywords are defined by reading through the filings to find word synonyms to the defined relations. For example, using keywords such as ”compete”, ”produce”, ”sell” to identify sentences containing those keywords which may contain the entities and relations.

Identify Entity Pairs

Once a key relation is identified in a sentence, the project uses dependency parsing to map out the sentence's structural hierarchy. It then uses NER to check if the subject and the object to the key relation contains entities. If the entities before and after the key relation match the predefined pattern, triples (h, r, t) are formed. Algorithm 2 for rule-based RE is presented below:

RE returns the following set of triple for Sample sentence 4:

[Linkedin] Acquire [Glint]

RE returns the following set of triples for Sample sentence 5:

[Microsoft] Has Competitor [BMC]

[Microsoft] Has Competitor [CA Technologies]

[Microsoft] Has Competitor [Hewlett-Packard]

[Microsoft] Has Competitor [IBM]

[Dynamics] Has Competitor [VMware]

RE returns the following set of triple for Sample sentence 6:

[Microsoft] Has Product [Bing]

[Microsoft] Has Product [Microsoft Advertising]

Algorithm 2 Rule-based RE algorithm

```
1:  procedure Relation Extraction($Sentence$)
2:      $TripleList$ =list()
3:      for Each $Word$ in $Sentence$ do
4:          if $Word == KeyRelationPatterns$ then
5:              $head$= Entities in $Word$'s subject
6:              $tail$= Entities in $Word$'s object
7:          end if
8:          if $head$ and $tail$ and $KeyRelation$ match tripple pattern then
9:              for $h$ in $head$ do
10:                 for $t$ in $tail$ do $TripleList$.append($h$,$KeyRelation$,$t$)
11:                 end for
12:             end for
13:         end if
14:     end for
15:     return $TripleList$
16: end procedure
```

4.6 Storing Triples

After triples are formed, graph database Neo4j is used to store the triples and form a KG. Neo4j can store the following information using its SQL-inspired query language Cypher:

- Node: represent the data records, which in this case is the entity.
- Relationship: connects Nodes with other Nodes, which in this case is the predefined relations from the ontology schema.
- Property: represents the value of each Node and Relationship. In this case, it is the name of the entities and relations.
- Label: categorizes the Nodes. In this case, it is the entity label.

This project uses Py2neo library, which integrates Cypher into python to allow access and control of the Neo4j database within a python application, to store each Node and their corresponding relations from the triples to Neo4j.

Algorithm 3 represents how the triples are stored and the sample result of a KG based on Sample sentence 4, 5, and 6 are stored in Neo4j in Figure 4.6.1. Green nodes represent company entities, and brown nodes represent product entities.

Algorithm 3 Storing KG algorithm

```
1: procedure Store the triples($TripleList$)
2:     for $triple$ in $TripleList$ do
3:         CREATE HeadNode(Entity, Entity Label)
4:         CREATE TailNode(Entity, Entity Label)
5:         CREATE RELATION(HeadNode, verb chunk, TailNode)
6:     end for
7: end procedure
```

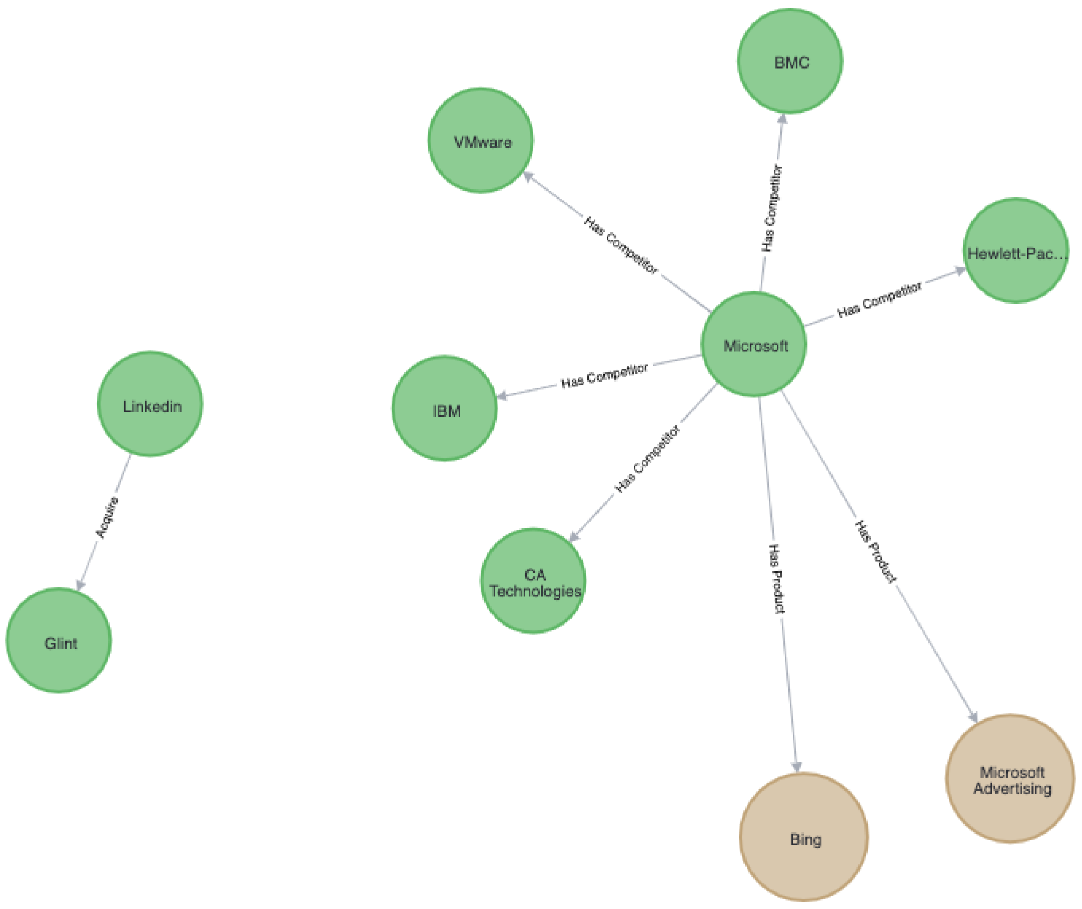


Figure 4.6.1: KG from sample sentences (Source: Author)

Chapter 5

Results and Discussion

This chapter describes the quantitative result of the company domain-specific KG built from SEC 10-K filings. The project uses ten 10-K filings to create rule-based patterns for the model and evaluate the solution's performance with another ten 10-K filings. Section 5.1 presents the results of the proposed solution and compares it with the open-source spaCy solution. Section 5.3 discusses the results.

5.1 KG Results

To evaluate the proposed model's performance, the project manually gathers all the available triples based on the ontology schema from each 10-K filing as a gold standard. The manually extracted triples are used to compare against the triples extracted using the proposed model described in Section 4. The performance of just using the spaCy model is also evaluated to check whether there is an improved performance using the proposed method. Furthermore, this project also compares the proposed model against the Azure text analytic model, which is trained using publicly listed companies from the is trained on publicly-listed companies to classify Company entities and classify Physical objects and computer objects as Product entities. The performance is evaluated using the Precision, Recall, and F1-score described in Section 3.6 The summary of the performance is in Table 5.1.1, Table 5.1.2 and Table 5.3.1. The following subsections look into the model's overall performance and the detailed performance of extracting each triple type.

Table 5.1.1: Results of the proposed model

	Recall	**Precision**	**F1 Score**
Competitor	67.2%	89.1%	76.6%
Acquire	64.5%	74.1%	69.0%
Products	39.2%	61.7%	47.9%
Overall	53.6%	75.7%	62.8%

Table 5.1.2: Just spaCy model Results

	Recall	**Precision**	**F1 Score**
Competitor	60.5%	91.0%	72.7%
Acquire	64.5%	69.0%	66.7%
Products	9.5%	22.6%	13.3%
Overall	36.7%	63.9%	46.7%

5.1.1 Overall

The average F1-score of the proposed model is 62.8%, which is higher than the F1 performance of 46.7% in the open-sourced model and the F1 performance of 58.8% in the Azure model.

The total Precision rate indicates that 75.7% percent of the triples extracted by the proposed model are correct, which is an 11.8% improvement compared to the open-source model. However, it is 1.1% lower than the Azure model. The total Recall rate indicates the model can retrieve 53.6% of the total triples in the ten 10-K filings compared to 36.7% using the open-source model and 47.6% using the Azure model.

5.1.2 Competitor Triples

The proposed model's F1-score on extracting competitor triples is 76.6%, which is higher than the open-sourced model's F1 performance of 72.7% and is lower than the Azure model's F1 performance of 78.8%.

The Precision rate indicates that 89.1% of the triples extracted by the proposed model are correct, which is 1.9% lower compared to the open-source model and 4.7% lower than the Azure model. The Recall indicates the model can retrieve 67.2% of the total

Table 5.1.3: Azure model Results

	Recall	**Precision**	**F1 Score**
Competitor	67.9%	93.8%	78.8%
Acquire	74.2%	76.8%	75.4%
Products	23.7%	52.3%	32.6%
Overall	47.6%	76.8%	58.8%

triples in the ten 10-K filings compared to 60.5% using the open-source model and 67.9% using the Azure model.

5.1.3 Acquire

The proposed model's F1-score on extracting company acquisition triple is 69.0%, which is higher than the open-sourced model's F1-score of 66.7% and is lower than the Azure model's F1 performance of 75.4%.

The Precision rate indicates that 74.1% of the triples extracted by the proposed model are correct, which is an 5.1% improvement compared to the open-source model. However, it is 2.7% lower than the Azure model. The Recall rate indicates the model can retrieve 64.5% of the total triples in the ten 10-K filings compared to 64.5% using the open-source model and 74.2% using the Azure model.

5.1.4 Product

The proposed model's F1 performance on extracting the company's product triples is 47.9%, which is higher than the open-sourced model's F1 performance of 13.3% and is higher than the Azure model's F1 performance of 32.6%.

The Precision rate indicates that 61.7% of the triples extracted by the proposed model are correct, which is an 39.1% improvement compared to the open-source model and 9.4% improvement compared to the Azure model. The Recall rate indicates the model can retrieve 39.2% of the total triples in the ten 10-K filings compared to 9.5% using the open-source model and 23.7% using the Azure model.

5.2 Reliability and Validity Analysis

Multiple acquisitions on the same testing data were made to ensure reliability in data acquisition. The same data was also run multiple times to ensure the reliability of the data results.

To calculate the performance of the proposed model using F1-score, Recall, and Precision, it requires all the available triples based on the ontology schema from the ten 10-K filings as a gold standard. To ensure the reliability of the gold standard triples, multiple manual gatherings of all the available triples based on each 10-K filing were done.

The data gathered in this project from the SEC EDGAR system are monitored at every stage of the development to ensure the validity of the data.

5.3 Discussion

5.3.1 Overall

The overall results show that the proposed model can improve the existing open-source model and retrieve company relations from the ten 10-K filings. However, the overall Recall performance is only around 50%, due to the low Recall performance in extracting Product triples.

5.3.2 Competitor Triples

Analysis of the extracted triples between the proposed model and the spaCy model shows the improvement is due to the improvement in reclassifying product entity from company entities, which is one of the limitations of the spaCy model mentioned in Section 4.5.4. The hybrid model can identify 90 unique Company type entities compared to 80 using just spaCy's NER.

However, the Azure model performs slightly better than the proposed model due to the increase in Precision. This is because Azure's NER is trained on company domain data.The Azure model can identify 91 unique Company type entities.

5.3.3 Acquisition Triples

The acquisition triples result shows that the proposed model has only a small margin of performance improvement. This is because there is only a slight NER improvement in identifying company entities in those sentences that contain acquisition triples in the ten 10-K reports.

The reason why the proposed model is not able to increase the performance significantly is due to the limitation of the proposed rule based NER. Since it can only identify a company entity that follows certain patterns, it cannot identify a company entity that does not fit the predefined patterns. Also, some of the entity types are ambiguous. For example, GitHub can be both a Company and a product. Therefore, the classification of those entity impacted the performance.

The Azure model performs slightly better than the proposed model both in both Precision and Recall. This is because Azure's NER is able to correctly identify company entities as the NER model is trained on publicly listed companies.

5.3.4 Product Triples

After analyzing the extracted triples, the project discovers the increase in performance is due to the increase in the total extracted company's product triples in the proposed model as the hybrid NER model can identify more product entities. The improvement mainly comes from the rule-based NER model in identifying new product entities and reclassifying wrongly labeled company entities to product entities. As a result, the Recall and Precision rate increase significantly. However, the Recall rate is below 50%.

Further analysis on why the Recall rate is still low concludes that the low Recall rate is because of two reasons:

- **Use of keywords in relation extraction:** The proposed relation extraction uses certain keywords to identify sentences where they may contain a company's competitor, acquisition, and product triples. However, keywords used to find a sentence containing company product relations are not as clear as the keywords used to find sentences containing company competitor relations and acquisition relations. Where the relation extraction can use "compete" or "acquire." The variety of keywords used to describe the company's products within a 10-K filing and each 10-K filings are highly influenced by the sector the company operates in, the type of product the company sells, and the writing style. Therefore, the keyword used to identify product triples from the training data cannot fully identify sentences containing the company's product triples.

- **Limit of Rule-based pattern to identify Product Entity:** Similar to the Acquisition Triples' performance, the rule-based NER is only able to identify Product type entity that follows certain patterns. Therefore, it is not able to identify product entities that do not fit the predefined patterns. Besides, some of the entity types are ambiguous. For example, Linkedin can be both a Company and a product.

This is evidenced by the high standard deviation of Product Triples' Precision and Recall performance of each 10-K report in Table 5.3.1.

Table 5.3.1: Results: Product Triples

	Precision	**Recall**	**F1 Score**
Report 1	40.0%	66.7%	50.0%
Report 2	7.1%	50.0%	12.5%
Report 3	58.8%	76.9%	66.7%
Report 4	50.0%	40.0%	44.4%
Report 5	66.7%	88.9%	76.2%
Report 6	33.3%	62.5%	43.5%
Report 7	16.7%	50.0%	25.0%
Report 8	40.9%	69.2%	51.4%
Report 9	75.0%	42.8%	54.4%
Report 10	50.0%	33.3%	40.0%
Stdev	**21.1%**	**17.7%**	**18.4%**

Chapter 6

Conclusions and Future Work

This chapter describes the conclusion by looking back to the purpose and goal of the project in Section 6.1 and reflects on the limitation of the project in Section 6.2. The future work and reflection on the project is presented in Section 6.3 and Section 6.4

6.1 Conclusions

With the development of NLP and graph database, it is now possible to extract structured information from an unstructured data source and store it. This can be applied to company domain information such as competitors, products, and merger and acquisition activities. By systematically extracting company domain information from financial text, and storing it in a graph database, time and cost reduction can be achieved. It also provides a better understanding of the structural relationships between companies which allows humans or machines to understand major risk factors and opportunities for a company. This project provides a model that can construct a company domain-specific KG that contains company-related entities and relationships from the SEC 10-K filings. The proposed model uses an open-source NLP tool combined with rule-based patterns in NER and RE, which eliminates the time-consuming data-labeling task in the statistical approach. By evaluating with ten 10-K filings, the model has the overall Recall of 53.6%, Precision of 75.7%, and the F1-score of 62.8%.

The result meets the project's goal of extracting company-related entities and relationships from SEC 10-K filings without data-labeling tasks and form and store the triples. However, the proposed method requires the time-consuming process of finding lexicon patterns from sentences to extract company-related entities and relationships. A further drawback is due to the limitation of the rule-based approach that cannot effectively extract triples if the entities and relations do not match the predefined patterns. This is evidenced by the Recall performance.

6.2 Limitations

There are two main limitations to this project. The first limitation is the lack of linguistic knowledge to effectively compose rule-based patterns, which led to a significant time analyzing the reports' lexicon patterns.

The second limitation is the lack of company domain-specific labeled data for triples to evaluate the proposed model. Due to the limited time and the significant time required to extract the triples manually, only ten reports were used to evaluate the proposed model's performance.

6.3 Future Work

The main areas for developing of future works are identified as followed:

Scalability

The next step for the project is to test how well the proposed model in terms of scalability by having more 10-K filings as an input with companies from different sectors. In addition to having more 10-K filings, another interesting area is to test how well the proposed model's performance with a different data source such as financial news.

Named Entities Recognition Performance Improvement

Many entities were identified by the noun chunk method used in the rule-based NER. However, many of those entities were still not classified by the predefined lexicon patterns and word vectors. Therefore, one potential solution is to use an unsupervised learning method of clustering to classify the unlabeled entity by its neighbor's entity type.

Furthermore, due to the limitation of the project mentioned in Chapter 1, another future work can be trying alternative NER model such as CRF and bootstrapping.

Expand the Relation Extraction

This project only focused on extracting 3 types of binary relations. An interesting area of work would be to increase the type of relations it can extract, such as is_supplier, is_customer, thus expanding the ontology schema.

In addition to expanding the ontology schema, another potential future working area can be expanding from sentence-level relation extraction to document level relation extraction. During the project, there has been much information at the document level that can form triples.

Relation Extraction Performance Improvement

As mentioned in Section 5.3.4, the use of keywords to find sentences containing key relation and the rule-based patterns to identify product entities cannot produce above 50% Recall. Therefore, another future work is to improve RE by using a statistical method to extract product relations.

6.4 Reflection

This project tries to extract structured company-related information from unstructured data systematically. This benefits organization that needs to map out company-related information by

- having a data storage of the retrieved company information to have a better understanding and control of the information
- allowing humans to work on more important decision-making tasks by replacing human labor on reading the long financial document
- having a systematic way to retrieve company information
- enhancing the transparency, readiness, and accessibility of company-related information

Furthermore, by mapping out company-related information, advanced applications can be developed to further increase the access of different companies and strengthen the economy.

Bibliography

[1] Ratajczak-Mrozek, Milena. “The Importance of Relationships and Embeddedness for Companies’ Internationalization and Performance—Results of Quantitative Study”. en. In: *Network Embeddedness: Examining the Effect on Business Performance and Internationalization*. Ed. by Milena Ratajczak-Mrozek. Palgrave Studies of Internationalization in Emerging Markets. Cham: Springer International Publishing, 2017, pp. 339–375. isbn: 978-3-319-56511-8. doi: 10.1007/978-3-319-56511-8_10. url: https://doi.org/10.1007/978-3-319-56511-8_10.

[2] Ford, David and Redwood, Michael. “Making sense of network dynamics through network pictures: A longitudinal case study”. en. In: *Industrial Marketing Management*. IMP Conference, Rotterdam, 2004 34.7 (Oct. 2005), pp. 648–657. issn: 0019-8501. doi: 10.1016/j.indmarman.2005.05.008. url: http://www.sciencedirect.com/science/article/pii/S0019850105000830 (visited on 01/10/2021).

[3] *Knowledge | Definition of Knowledge by Oxford Dictionary on Lexico.com also meaning of Knowledge*. en. url: https://www.lexico.com/definition/knowledge (visited on 01/10/2021).

[4] Lieberman, David A. *Learning and Memory*. Cambridge University Press, 2011. doi: 10.1017/CBO9781139046978.

[5] Rodrigues, Mário and Teixeira, António. *Advanced Applications of Natural Language Processing for Performing Information Extraction*. en. SpringerBriefs in Electrical and Computer Engineering. Cham: Springer International Publishing, 2015. isbn: 978-3-319-15562-3 978-3-319-15563-0. doi: 10.1007/978-3-319-15563-0.

url: http://link.springer.com/10.1007/978-3-319-15563-0 (visited on 03/10/2020).

[6] Piskorski, Jakub and Yangarber, Roman. “Information Extraction: Past, Present and Future”. en. In: *Multi-source, Multilingual Information Extraction and Summarization*. Ed. by Thierry Poibeau, Horacio Saggion, Jakub Piskorski, and Roman Yangarber. Theory and Applications of Natural Language Processing. Berlin, Heidelberg: Springer, 2013, pp. 23–49. isbn: 978-3-642-28569-1. doi: 10.1007/978-3-642-28569-1_2. url: https://doi.org/10.1007/978-3-642-28569-1_2.

[7] Bollacker, Kurt, Evans, Colin, Paritosh, Praveen, Sturge, Tim, and Taylor, Jamie. “Freebase: a collaboratively created graph database for structuring human knowledge”. In: *In SIGMOD Conference*. 2008, pp. 1247–1250.

[8] S, Patil N., P, Kiran, P, Kiran N., and M, Naresh Patel K. “A Survey on Graph Database Management Techniques for Huge Unstructured Data”. en. In: *International Journal of Electrical and Computer Engineering (IJECE)* 8.2 (Apr. 2018). Number: 2, pp. 1140–1149. issn: 2722-2578. doi: 10.11591/ijece.v8i2.pp1140-1149. url: http://ijece.iaescore.com/index.php/IJECE/article/view/10417 (visited on 11/10/2020).

[9] Noel, Steven, Harley, E., Tam, K.H., Limiero, M., and Share, M. “CyGraph: Graph-Based Analytics and Visualization for Cybersecurity”. In: Jan. 2016. doi: 10.1016/bs.host.2016.07.001.

[10] Kejriwal, Mayank. *Domain-Specific Knowledge Graph Construction*. Springer International Publishing, 2019. isbn: 978-3-030-12374-1. doi: 10.1007/978-3-030-12375-8. url: http://link.springer.com/10.1007/978-3-030-12375-8.

[11] Ruan, Tong, Xue, Lijuan, Wang, Haofen, Hu, Fanghuai, Zhao, Liang, and Ding, Jun. “Building and Exploring an Enterprise Knowledge Graph for Investment Analysis”. en. In: *The Semantic Web – ISWC 2016*. Lecture Notes in Computer Science. Springer, Cham, Oct. 2016, pp. 418–436. isbn: 978-3-319-46546-3. doi: 10.1007/978-3-319-46547-0_35. url: https://link-springer-com.focus.lib.kth.se/chapter/10.1007/978-3-319-46547-0_35.

[12] Liu, Yang, Zeng, Qingguo, Ordieres Meré, Joaquín, and Yang, Huanrui. *Anticipating Stock Market of the Renowned Companies: A Knowledge Graph*

Approach. en. 2019. doi: 10.1155/2019/9202457. url: https://www.hindawi.com/journals/complexity/2019/9202457/.

[13] Liu, Jue, Lu, Zhuocheng, and Du, Wei. “Combining Enterprise Knowledge Graph and News Sentiment Analysis for Stock Price Prediction”. eng. In: Jan. 2019. isbn: 978-0-9981331-2-6. doi: 10.24251/HICSS.2019.153. url: http://scholarspace.manoa.hawaii.edu/handle/10125/59565.

[14] *Open Working Group, Sustainable Development Goals, Development Goals, and Sustainable Development Goals. Sustainable Development Goals and targets*. EN. 2015. url: https://sustainabledevelopment.un.org/focussdgs.html.

[15] Håkansson, Anne. “Portal of Research Methods and Methodologies for Research Projects and Degree Projects”. en. In: *Computer Engineering* (2013), p. 7.

[16] *SEC.gov | About the SEC*. url: https://www.sec.gov/about.shtml (visited on 05/06/2020).

[17] *msft-10k_20190630.htm*. url: https://www.sec.gov/Archives/edgar/data/789019/000156459019027952/msft-10k_20190630.htm (visited on 11/09/2020).

[18] Allen, James F. “Natural Language Processing”. en. In: *Encyclopedia of Cognitive Science*. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470018860.s00078. American Cancer Society, 2006. isbn: 978-0-470-01886-6. doi: 10.1002/0470018860.s00078. url: http://onlinelibrary.wiley.com/doi/abs/10.1002/0470018860.s00078 (visited on 05/06/2020).

[19] Jurafsky, Dan. *Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition*. eng. 2. ed.. Prentice Hall series in artificial intelligence. Upper Saddle River, N.J.: Pearson Education International/Prentice Hall, 2009. isbn: 978-0-13-504196-3.

[20] Yang, Chin-Sheng and Shih, Hsiao-Ping. “A Rule-Based Approach For Effective Sentiment Analysis”. en. In: (), p. 9.

[21] *Introducing the Knowledge Graph: things, not strings*. en. Library Catalog: googleblog.blogspot.com. url: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html (visited on 05/06/2020).

[22] Kiss, Tibor and Strunk, Jan. “Unsupervised Multilingual Sentence Boundary Detection”. en. In: *Computational Linguistics* 32.4 (Dec. 2006), pp. 485–525. issn: 0891-2017, 1530-9312. doi: `10.1162/coli.2006.32.4.485`. url: `http://www.mitpressjournals.org/doi/10.1162/coli.2006.32.4.485` (visited on 06/08/2020).

[23] Chomsky, Noam. *Syntactic Structures*. en. Publication Title: Syntactic Structures. De Gruyter Mouton, Sept. 2009. isbn: 978-3-11-021832-9. url: `https://www.degruyter.com/view/title/14418` (visited on 06/08/2020).

[24] Song, Dezhao, Schilder, Frank, Hertz, Shai, Saltini, Giuseppe, Smiley, Charese, Nivarthi, Phani, and Hazai, Oren. “Building and Querying an Enterprise Knowledge Graph”. In: *IEEE Transactions on Services Computing* 12.3 (2019), pp. 356–369. issn: 2372-0204. doi: `10.1109/TSC.2017.2711600`.

[25] Bizer, Christian, Lehmann, Jens, Kobilarov, Georgi, Auer, Sören, Becker, Christian, Cyganiak, Richard, and Hellmann, Sebastian. “DBpedia - A crystallization point for the Web of Data”. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 7.3 (Sept. 2009), pp. 154–165. issn: 1570-8268. doi: `10.1016/j.websem.2009.07.002`. url: `https://doi.org/10.1016/j.websem.2009.07.002` (visited on 06/08/2020).

[26] Sarawagi, Sunita. “Information Extraction”. English. In: *Foundations and Trends® in Databases* 1.3 (Nov. 2008). Publisher: Now Publishers, Inc., pp. 261–377. issn: 1931-7883, 1931-7891. doi: `10.1561/1900000003`. url: `https://www.nowpublishers.com/article/Details/DBS-003` (visited on 05/07/2020).

[27] Etzioni, Oren, Banko, Michele, Soderland, Stephen, and Weld, Daniel S. “Open information extraction from the web”. en. In: *Communications of the ACM* 51.12 (Dec. 2008), pp. 68–74. issn: 0001-0782, 1557-7317. doi: `10.1145/1409360.1409378`. url: `https://dl.acm.org/doi/10.1145/1409360.1409378` (visited on 05/06/2020).

[28] Schmitz, Michael, Bart, Robert, Soderland, Stephen, and Etzioni, Oren. “Open Language Learning for Information Extraction”. en. In: (), p. 12.

[29] Levesque, H J. “Knowledge Representation and Reasoning”. In: *Annual Review of Computer Science* 1.1 (1986). _eprint: https://doi.org/10.1146/annurev.cs.01.060186.001351, pp. 255–287. doi: `10.1146/annurev.cs.01.060186.001351`. url: `https://doi.org/10.1146/annurev.cs.01.060186.001351` (visited on 06/07/2020).

[30] “Introduction”. en. In: *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Ed. by Michel Chein and Marie-Laure Mugnier. Advanced Information and Knowledge Processing. London: Springer, 2009, pp. 1–17. isbn: 978-1-84800-286-9. doi: `10.1007/978-1-84800-286-9_1`. url: `https://doi.org/10.1007/978-1-84800-286-9_1` (visited on 06/08/2020).

[31] Ehrlinger, Lisa and Wöß, Wolfram. “Towards a Definition of Knowledge Graphs”. en. In: (), p. 4.

[32] Nadeau, David and Sekine, Satoshi. “A survey of named entity recognition and classification”. en. In: (), p. 20.

[33] Riloff, Ellen and Jones, Rosie. “Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping”. en. In: (), p. 6.

[34] Schön, Saskia, Mironova, Veselina, Gabryszak, Aleksandra, and Hennig, Leonhard. “A Corpus Study and Annotation Schema for Named Entity Recognition and Relation Extraction of Business Products”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. url: `https://www.aclweb.org/anthology/L18-1704` (visited on 07/22/2020).

[35] Mikheev, Andrei, Grover, Claire, and Moens, Marc. “Description of the LTG system used for MUC-7”. English. In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998* (1998). Publisher: Association for Computational Linguistics. url: `https://www.research.ed.ac.uk/portal/en/publications/description-of-the-ltg-system-used-for-muc7(0595b963-47d6-4958-8528-8a443c5fa56e).html` (visited on 11/09/2020).

[36] Rocktäschel, Tim, Weidlich, Michael, and Leser, Ulf. “ChemSpot: a hybrid system for chemical named entity recognition”. en. In: *Bioinformatics* 28.12 (June 2012). Publisher: Oxford Academic, pp. 1633–1640. issn: 1367-4803. doi: 10.1093/bioinformatics/bts183. url: https://academic.oup.com/bioinformatics/article/28/12/1633/266861 (visited on 07/22/2020).

[37] Stewart, Michael, Enkhsaikhan, Majigsuren, and Liu, Wei. “ICDM 2019 Knowledge Graph Contest: Team UWA”. In: *arXiv:1909.01807 [cs]* (Sept. 2019). arXiv: 1909.01807. url: http://arxiv.org/abs/1909.01807 (visited on 06/08/2020).

[38] Rubens, Matthew and Agarwal, Puneet. “Information Extraction from Online Automotive Classifieds”. en. In: (), p. 8.

[39] Katharina Sienčnik, Scharolta. “Adapting word2vec to Named Entity Recognition”. In: *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*. Vilnius, Lithuania: Linköping University Electronic Press, Sweden, May 2015, pp. 239–243. url: https://www.aclweb.org/anthology/W15-1830 (visited on 08/01/2020).

[40] Wirth, Rüdiger and Hipp, Jochen. “CRISP-DM: Towards a Standard Process Model for Data Mining”. en. In: (), p. 11.

[41] Ashraf, Rasha. *Scraping EDGAR With Python*. en. SSRN Scholarly Paper ID 3230156. Rochester, NY: Social Science Research Network, June 2017. url: https://papers.ssrn.com/abstract=3230156 (visited on 03/11/2020).

[42] Farmakiotou, Dimitra, Karkaletsis, V., Koutsias, J., Sigletos, George, Spyropoulos, C., and Stamatopoulos, P. *RULE-BASED NAMED ENTITY RECOGNITION FOR GREEK FINANCIAL TEXTS*. en. 2000. url: /paper/RULE-BASED-NAMED-ENTITY-RECOGNITION-FOR-GREEK-TEXTS-Farmakiotou-Karkaletsis/779126e98eacf61314abf08b2c7dad997a79cfa5 (visited on 11/08/2020).

[43] Jiang, Ridong, Banchs, Rafael E., and Li, Haizhou. “Evaluating and Combining Name Entity Recognition Systems”. In: *Proceedings of the Sixth Named Entity Workshop*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 21–27. doi: 10.18653/v1/W16-2703. url: https://www.aclweb.org/anthology/W16-2703 (visited on 06/08/2020).

[44] Manning, Christopher, Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven, and McClosky, David. "The Stanford CoreNLP Natural Language Processing Toolkit". en. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 55–60. doi: `10.3115/v1/P14-5010`. url: `http://aclweb.org/anthology/P14-5010` (visited on 06/08/2020).

[45] *Facts & Figures · spaCy Usage Documentation*. en. Library Catalog: spacy.io. url: `https://spacy.io/` (visited on 06/08/2020).

[46] *What Is a Graph Database and Property Graph | Neo4j*. en. Library Catalog: neo4j.com. url: `https://neo4j.com/developer/graph-database/` (visited on 06/08/2020).

[47] *Cayley*. en-us. Library Catalog: cayley.io. url: `//` (visited on 06/08/2020).

[48] Makhoul, John, Kubala, Francis, Schwartz, Richard, and Weischedel, Ralph. "Performance Measures For Information Extraction". In: *In Proceedings of DARPA Broadcast News Workshop*. 1999, pp. 249–252.

[49] *File:Precisionrecall.svg*. en. url: `https://en.wikipedia.org/wiki/File:Precisionrecall.svg` (visited on 01/24/2021).

TRITA-EECS-EX-2021:21