Title
How to Identify Who Is the Chief Executive Officer in the Unconstructed Financial Reports - Use the Proxy Statements from Tesla as an Example

Author
CHEN, YUAN-YI

Publication Date
2018

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

How to Identify Who Is the Chief Executive Officer in the Unconstructed Financial Reports - Use the Proxy Statements from Tesla as an Example

A thesis submitted in partial satisfaction
of the requirements for the degree Master of Science
in Applied Statistics

by

Yuan-Yi Chen

2018

ABSTRACT OF THE THESIS

How to Identify Who Is the Chief Executive Officer in the Unconstructed Financial Reports - Use the Proxy Statements from Tesla as an Example

by

Yuan-Yi Chen

Master of Science in Applied Statistics
University of California, Los Angeles, 2018
Professor Ying Nian Wu, Chair

According to M Firth, PMY Fung, OM Rui (2006), the stock market is pretty sensitive to the top management turnover. Therefore, it is important to monitor the top management turnover in a very short time. This thesis attempts to use statistical and machine learning techniques to identify the Chief Executive Officer in a specific company. The natural language processing techniques we used may be able to show who is the Chief Executive Officer from millions of words in financial reports within few minutes. The whole process comes with four sectors: data collection, manipulation, named entity recognition and relationship analysis. We demonstrate this method with Tesla's proxy statements (code: DEF 14A) from the U.S. Securities and Exchange Commission. The output shows that we can use both the named entity recognition algorithm with word2vec algorithm to detect the relationship between a job title and a human name.

The thesis of Yuan-Yi Chen is approved.

Nicolas Christou

Hongquan Xu

Ying Nian Wu, Committee Chair

University of California, Los Angeles

2018

Acknowledgments

I am deeply thankful for the helps from my mother, brother and my father. This Master's thesis cannot be finished without their supports. I would like to express my thankfulness to my thesis committees, Professor Ying Nian Wu, Professor Nicolas Christou and Professor Hongquan Xu for reviewing my thesis and guiding me toward my first research in natural language processing. I do appreciate the helps from our Statistics Department, especially the Student Affair Officer Leyden Laurie for offering me lots of administrative supports. Last but not the least, I'd deeply appreciate my life mentor – Professor Vivian Lew, for caring me and accompanying me confront the darkness.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

Named entity recognition has been widely used in several area since 1991. According to David Nadeau, Satoshi Sekine (2007), it was used to extract "proper names" like human names or company names (Lisa F. Rau (1991)) at the very beginning. Gradually, it became popular to detect "locations" (M. Fleischman 2001, S. Lee & Geunbae Lee 2005) and "miscellaneous" contents like email addresses (D. Maynard et al. 2001), book titles (J. Zhu et al. 2005) and so on. As the machine learning methodology shows its computability and predictability, researchers can now combine it with named entity recognition algorithms to detect a relationship between an object and an identity.

Due to the development of Internet-related applications, the scale of data grows exponentially, which brings both advantages and disadvantages. Although we can gain more data to monitor the performance of a company, it takes a reader lots of time to scrutinize all information.

There are many researches analyzing what factors may influence the stock market (F Wang, Y Xu (2004)). According to Chapple, L. & Humphrey, J.E. J (2014), *"we find some weak evidence of a negative correlation between having multiple women on the board and performance"* (as cited in Journal of Business Ethics 2014), which identifies the top managements can be a factor that impacts the stock market. This is the starting point of why we focus on finding the name of the Chief Executive Officer.

My thesis focuses on pairing a job title with a human name within a company's proxy statements. To be more specific, people do not need to read through all company reports in order to know who is the Chief Executive Officer in a specific company.

After collecting and cleaning the proxy statements we got from the U.S. Securities and Exchange Commission (SEC), we split, tokenized, and applied the named entity extraction algorithm to the clean data in order to extract a list of human names. Several miscellaneous steps like removing middle names, concatenating first names and last names, removing spaces were finished at this moment.

Afterward, word2vec algorithm was applied to the new data set, which the original full names were replaced by the concatenated names we made from the previous steps. We finally got a list of similarities that shows the relationship between a human name and the job title "Chief Executive Officer". Comparing with the average of human reading speed, it is an achievement of how natural language processing can help gain a faster insight in the financial industry.

1.2 Problems

After we define the direction of our research, here comes the second problem - how can we monitor this issue? Also, there are tons of news, reports for both investors and governments to read. From which data source can we rely on?

Fortunately, according to the security laws in United States, companies established in the United States which listed at a stock exchange market need to report their situations to the U.S. Securities and Exchange Commission (SEC). That is to say, we can narrow down the scope of data analysis to those formal reports that a company provided.

Besides the difficulty of data sources, there are also several technical difficulties to cope with. Firstly, we may need to extract human names in order to identify who are the top managers in the company. We also need to match the human names we collected with the job title "Chief Executive Officer". Sadly speaking, there are many researches focusing on named entity

recognition like NLTK, Stanford CoreNLP. However, there are quite few researches evaluating the relationship between a human name and a job title. This problem is what my thesis aim to address.

Chapter 2
Data Source

The central question in this thesis is to find the name of Chief Executive Officer, which usually appears in several resources. We may need to filter the data sources before we clean the data. For the purpose of identifying the most reliable data source, we would like to classify the typical financial data into four kinds of resources:

Type	**Definition**	**Examples**
Traditional Media	Usually operates by companies. Contents are more objective.	News from TV channels, newspapers, magazines.
Social Media	User-generated contents. Contents are more subjective	Facebook, Twitter, Blogs
Accounting Statements	Financial Statements provided by the accounting agency	Balance sheet, income statement
Government Reports	The reports that Listed companies provide to the U.S. Securities and Exchange Commission (SEC)	Annual report, power of attorney, proxy statement

Table 1: Four types of data sources

Concerning our topic - Identify who is the Chief Executive Officer, we need to have at least two things in our data source. The first one is the object. In this thesis, we assume it as the name of

the Chief Executive Officer. The other one is the identity, which means the job title (i.e. Chief Executive Officer).

For the purpose of measuring the relationship between an object and an identity, we determine to use the fourth data source, which is the reports that each listed companies need to report to the U.S. Securities and Exchange Commission (SEC).

There are some factors which influence our selection. Firstly, the traditional media usually wants to attract audiences in a few second. That is to say, they tend to show a table, a dashboard, a chart instead of full-scope data, which may not be adequate to apply in our research.

As for the social media, even though some users will post the full-scope data set, usually, the data is not in a consistent format. For example, users may only post a certain year data instead of the data sets with whole year data involved. This makes us hard to monitor top management transitions by year.

For the third source, although the financial statements have the consistent, full-scope data and can also fulfill our requirements (i.e. have both objects and identities), I still choose not to use them because the financial statements often disclose cash-related stuffs within the reports. Considering the "word2vec" algorithm we may apply later, we hope to have text contents come from different aspects instead of just cash-related contents.

Comparing to above non-selected data sources, the reports which listed companies turn to the U.S. Securities and Exchange Commission (SEC) has the following advantages:

A. Reliable Sources

According to the law, it is mandatory for every listed company to hand in different kinds of reports to the U.S. Securities and Exchange Commission (SEC). In order to not violate the law, the data which listed company provides is much more formal and reliable.

B. Consistent Contents

In order to compare the performance of last year, listed company usually discloses both the latest data and data within two years. For example, most of the proxy statement reports show the 3-years compensation tables like Table 2.

Name and Principal Position	Year	Salary ($)
Elon Musk	2017	49,920
Chief Executive Officer and Chairman	2016	45,936
	2015	37,584
Jeffrey B. Straubel	2017	249,600
Chief Technology Officer	2016	250,560
	2015	250,560
Deepak Ahuja(3)	2017	428,846
Chief Financial Officer	2015	339,300
Doug Field	2017	300,000
Senior Vice President, Engineering	2016	301,153
	2015	306,923
Jason Wheeler	2017	174,041
Former Chief Financial Officer	2016	501,931
	2015	46,154
Jon McNeill	2017	500,000
Former President, Global Sales and Service	2016	501,923

Table 2: 2018 Compensation Table from Tesla Inc.

C. Broad Scopes

Not only financial related contents (like an operating cost, tax) but also the demographic data like a biography/gender will be listed within reports. In comparison with the financial statements, these reports provide much more corpus for us to apply the natural language processing algorithms.

D. Diversified Reports

There are roughly 158 types of reports provided by each listed company. For example, annual reports, power of attorney reports, proxy statement reports. (Provided by the database from the U.S. Securities and Exchange Commission (SEC): https://www.sec.gov/forms)

Although we have defined which data source to use, it will be too cumbersome if we apply algorithms to all 158 types of reports for each company. Therefore, we mainly focus on one company “Tesla” and its proxy statement (filetype: DEF 14A) from year 2011 to 2018.

Chapter 3
Thesis Structure

This thesis involves a lot of data cleansing processes. In Chapter 4, we will introduce the methodology of how we extract human names (i.e. Stanford NER) and analyze the relationship between words (i.e. word2vec). On the other hand, Chapter 5 will go deep to the details about how we apply those methods to the data we cleaned and how could we measure the performance of those natural language processing algorithms. The Figure 1 shows the whole structure of both the data manipulation and analysis.

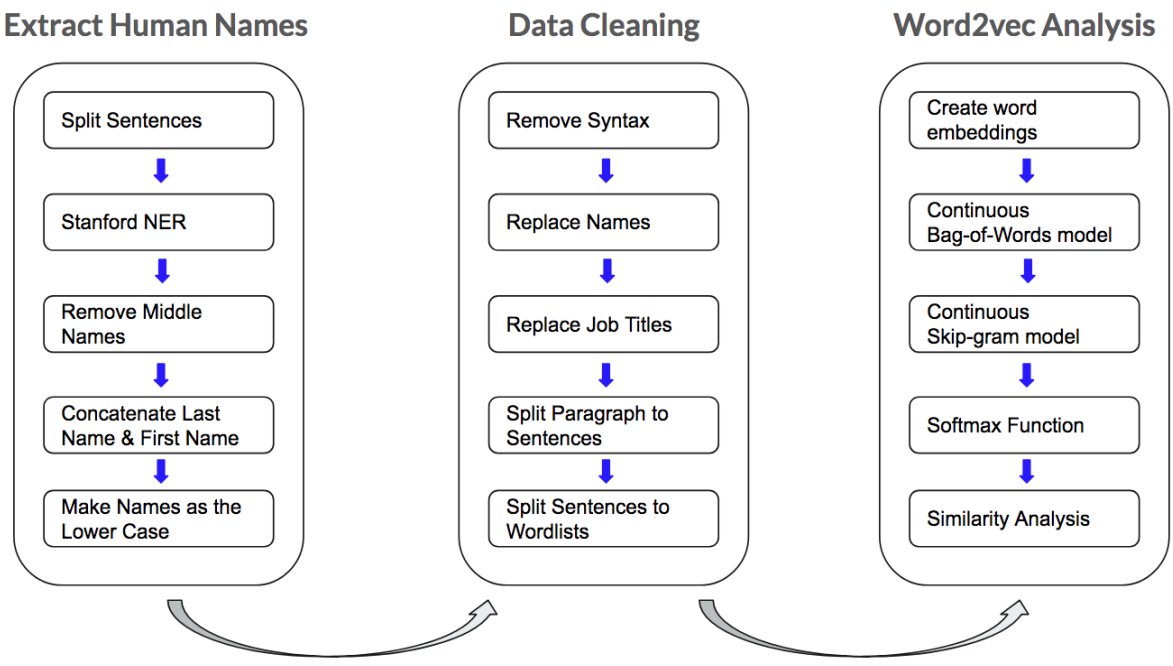

Figure 1: Structure of the Data Manipulation

Chapter 4
Methodology

The main idea of this thesis is to measure the relationship between an object (i.e. human name) and an identity (i.e. job title). Therefore, our assumption is, the much closer relationship between an object and an identity, the higher probability that this person owns that job title. In order to approach to the answer, we use "Word2vec" to measure the distance between each word vector and name this relationship as the "similarity". Furthermore, we also need to extract human names before applying word2vec. This is why we not only use "Word2vec" but also apply other natural language processing packages like "NLTK" and "Stanford Named Entity Recognition" to texts.

Our assumption is made based on the special structured in proxy statement reports. We found that the "name" of the Chief Executive Officer often comes before or after the "job title". That is to say, even though we may have hundreds of human names in a proxy statement report, we can identify who is the Chief Executive Officer by examining which name is closer to the "Chief Executive Officer". This finding strengthens our assumption that word2vec may work to measure a relationship between a human name and a job title because it evaluates the "similarity" between words. The following example 1 can show an overview of how it works with "Word2vec".

Example 1 (*From 2018 Proxy Statement page 38)**:**
Tesla is committed to fair and competitive compensation for its employees. Moreover, **Elon Musk**, our **Chief Executive Officer**, has agreed to a compensation arrangement in the 2018 CEO Performance Award that is substantially tied to the appreciation of our market capitalization.

The above example shows that we have two named entities: Tesla and Elon Musk. Also, we have our target job title "Chief Executive Officer". Word2vec algorithm can evaluate the similarities of the following two pairs (i.e. an object versus an identity):

A. Tesla and Chief Executive Officer
B. Elon Musk and Chief Executive Officer

Therefore, we may be able to identify who is the Chief Executive Officer by finding in which pair does it have the highest similarity.

Before we really go deep into the whole analysis, we would like to introduce how we combine three algorithms - NLTK, Stanford NER and Word2vec together to approach our assumption. We classify our approaches into three steps:

A. Split Contents - NLTK
B. Extract Human Names – Stanford NER
C. Analyzing the similarities between names and job titles – Word2vec

4.1 Split Contents - NLTK

According to D Nadeau, S Sekine (2007), we can classify corpus into seven word-level features: Cases (i.e. capital letters), punctuations, digits, characters, morphologies, part-of-speeches (i.e. names, verb, noun etc) and functions. Also, there are tons of variations of each word like abbreviations, affixes, derived words and so on. In this situation, extracting specifically "human names" becomes difficult.

We roughly get 30,000 words within a single proxy statement (DEF 14A) report from Tesla Inc. Our first step is to split each paragraph to sentences and each sentences to words. The package "NLTK" provides us an useful tool to achieve this step.

NLTK is the abbreviation of **Natural Language Toolkit**, which is a Python library for the natural language processing. From NLTK 3.3 documentation, we know that it has the power to split, tokenize, stem, tag, parse the texts. It can also demonstrate the parse tree which shows the structure of a given sentence like the Figure 2.

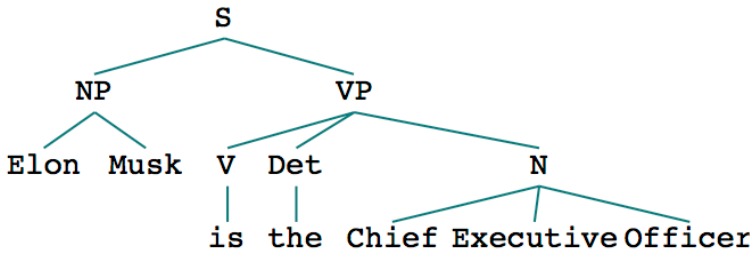

Figure 2: Parse tree

In order to extract human names in the next step, we rely on "NLTK" to split sentences to words for further applying the Stanford Named Entity Recognition algorithm. Traditionally, there are many methods to split sentences to words. Assume the sentence we want to split is called "sentt", based on the outputs, we can classify the splitting functions into two methods:

A. Keep the punctuations: e.g. "NLTK"
B. Remove the punctuations: e.g. "split(sentt)" or "re.split('\W', sentt)"

Generally speaking, we will ignore any punctuation in the parse tree because it may not impact the text structure. However, in order to extract full names in the later steps, we need to keep both the stopping words (like "the", "a", "at") and punctuations. Comparing to other functions like "split()", which will ignore punctuations, "NLTK" retains punctuations while chopping sentences into pieces. Therefore, "NLTK" is our best choice in this case.

	Output
Original	Elon Musk, our Chief Executive Officer, has agreed to a compensation arrangement in the 2018 CEO Performance Award
Using "NLTK" (Remain the comma)	'Elon', 'Musk', ',', 'our', 'Chief', 'Executive', 'Officer', ',', 'has', 'agreed', 'to', 'a', 'compensation', 'arrangement', 'in', 'the', '2018', 'CEO', 'Performance', 'Award'
Using "split()" (Delete the comma)	'Elon', 'Musk,', 'our', 'Chief', 'Executive', 'Officer,', 'has', 'agreed', 'to', 'a', 'compensation', 'arrangement', 'in', 'the', '2018', 'CEO', 'Performance', 'Award'

Table 3: Splitting sentence using NLTK & split function

4.2 Extracting Human Names – Stanford NER

Secondly, we apply Stanford Named Entity Recognition (Stanford NER) algorithm to the list of words we got from the previous step in order to get pieces of names. Unlike other natural language processing algorithm, which will label each words with multi word expressions (see Figure 2: parse tree), Stanford Named Entity Recognition will only tag words with two kinds of labels: Object or Person. Considering we only need to extract human names in this step, we choose Stanford Named Entity Recognition as our algorithm instead of using "NLTK".

	Output
Original	Elon Musk, our Chief Executive Officer, has agreed to a compensation arrangement in the 2018 CEO Performance Award
Using "NLTK"	Elon Musk , our Chief Executive Officer , NNP NNP , PRP NNP NNP NNP , has agreed to a compensation arrangement in VBZ VBZ TO DT NN NN IN the 2018 CEO Performance Award DT CD NNP NNP NNP
Using Stanford NER	Elon Musk , our Chief Executive Officer , **Person Person** O O O O O O has agreed to a compensation arrangement in O O O O O O O the 2018 CEO Performance Award O O O O O

Table 4: Different Tagging methods

However, the story has not yet finished, as we found that Stanford Named Entity Recognition algorithm has some minor disadvantages that need to be modified.

A. Split one name into first name, middle name and last name

The original Stanford NER will examine each word and classify those words into two types (i.e. Object or Person). Since we have already split a sentence to list of words, either a first name or a middle name, or even a last name will be regarded as a word. This finally caused a human name to be split into several pieces of names like what Table 5 shows.

	Output
Original	Moreover, Elon Musk, our Chief Executive Officer, has agreed to a compensation arrangement in the 2018 CEO Performance Award that is substantially tied to the appreciation of our market capitalization.
Before Modification	1st name: Elon 2nd name: Musk
After Modification	1st name: Elon Musk

Table 5: Named Entity Recognition with Modification

Since our original goal is to split a person's "full name", we need to modify the Stanford Named Entity Recognition algorithm in order to extract full names. Our strategy here is to combine pieces of names into full names if there are groups of "Person" tags. That is to say, if there are two words, which have tags "Person" side by side. We will concatenate these two words to one word.

B. Names concatenated together

Due to the unstructured text structure in every proxy statement report, we often encounter a problem that two full names will be concatenated together because of:

- The above modification (Section 4-2-A)
- Ignore the punctuations (has been solved in Section 4-1)
- HTML type of document, which has several spaces between names like Figure 3 shows

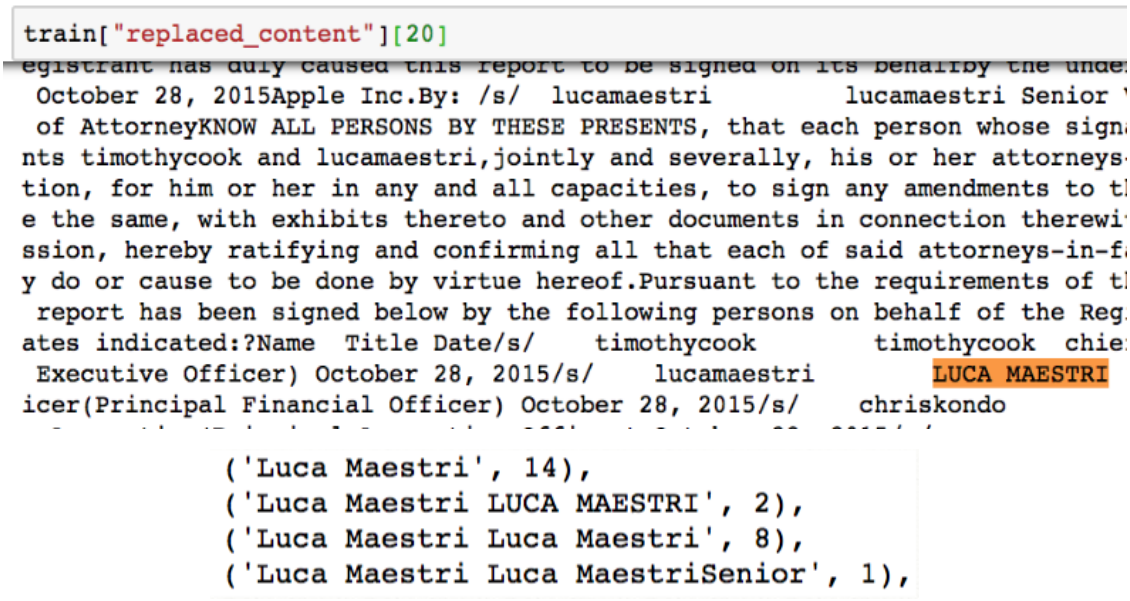

Figure 3: Space Problem

These problems can be solved by several data cleansing steps like replacing a space with a comma. After several data cleaning steps, we can have a list of human names from each proxy statement report like Table 6 shows.

	Output
Original	The persons named as proxy holders, Elon Musk, Deepak Ahuja and Todd Maron, or any of them, will have discretion to vote the proxies held by them on those matters in accordance with their best judgment.
Using split() & Stanford NER (Missed one human name)	1st name: Deepak Ahuja 2nd name: Todd Maron
Using NLTK & Stanford NER	1st name: Elon Musk 2nd name: Deepak Ahuja 3rd name: Todd Maron

Table 6: Different Named Entity Recognition Methods

4.3 Analyzing the relationship between names and job titles - Word2vec

We have plenty of ways to measure the relationship between objects and names. For example, the term frequency (Hans Peter Luhn (1957)), uses a weighting factor to evaluate how important is a word in a document. Also, we have the inverse document frequency (Karen Spärck Jones (1972)), to get rid of words which appear very often but indeed do not contribute much more meaning as its frequency increases. The thing is, those algorithms usually take the frequency into consideration. Since we found the special structure of the proxy statement, which the names usually followed by corresponding job titles, we can leverage this attribute to measure the distance between an object and an identity.

A proxy statement report, is a report that a company discloses voting ballot, board of directors and decisions about matters to their stakeholders. To be specific, the name of the Chief Executive Officer is usually placed within the board of directors or executive officers' sections. Generally speaking, their full names will be placed near the job titles. This attribute enable us to apply "word2vec" to measure the distance between an object and a name, as it takes the coefficient of a shallow neural network to show us the similarity.

Word2vec (Tomas Mikolov (2013)), takes a large amount of corpus of texts to make word embeddings and predicts the possibility of a human name appears within a surrounding window of a job title. According to the research from Tomas Mikolov (2013), each word will be identified as a vector and placed in a vector space. Afterward, those vectors will be used in the following two steps: continuous bag-of-words (CBOW) and continuous skip-gram.

In order to apply the above algorithms, we will concatenate both names and the job title. Table 7 is an example of how a sentence looks like in the continuous bag-of-words (CBOW) and continuous skip-gram models.

	Output
Original Sentence	Elon Musk is the Chief Executive Officer who wants to take Tesla private.
After the Concatenation	ElonMusk is the ChiefExecutiveOfficer who wants to take Tesla private.
Continuous Bag-of-Words (3 Grams)	1st: "ElonMusk is the" 2nd: "is the ChiefExecutiveOfficer" 3rd: "the ChiefExecutiveOfficer who" 4th: "ChiefExecutiveOfficer who wants" 5th: "who wants to" 6th: "wants to take" 7th: "to take Tesla" 8th: "take Tesla private"
Continuous Skip-Gram (Training samples)	1st: {"ElonMusk is the", "ElonMusk is ChiefExecutiveOfficer"} 2nd: {"is the ChiefExecutiveOfficer", "is the who"} 3rd: {"the ChiefExecutiveOfficer who", "the ChiefExecutiveOfficer wants"} 4th: {"ChiefExecutiveOfficer who wants", "ChiefExecutiveOfficer who to"} 5th: {"who wants to", "who wants take"} 6th: {"wants to take", "wants to Tesla"} 7th: {"to take Tesla", "to take private"} 8th: {"take Tesla private"}

Table 7: How a Sentence Looks Like within the Continuous Bag-of-Words and Continuous Skip-Gram

As an user identifies the context window, this algorithm will measure the possibility of a human name which falls within the context window. In our case, as we plug in the name of Chief Executive Officer within the context window (length equals to three). Word2vec can estimate the possibility of the name "ElonMusk" that falls nearby the job title "ChiefExecutiveOfficer". Thus, we can figure out the relationship between a job title and a name by measuring how close is these two objects.

$$Prob\ (human\ name \mid job\ title)$$

Since we may have several human names in a report, we will have many possibility index of one human name given by a job title. In order to reduce the bias, we will use the softmax function (Bishop, Christopher M. (2006)) in order to get the final probability.

$$\text{Softmax} = \frac{exp\ (\upsilon_{job\ title} \cdot \upsilon_{name})}{\sum_{k \in V} exp\ (\upsilon_{job\ title} \cdot \upsilon_{k})} = P(\upsilon_{name} \mid \upsilon_{job\ title})$$

K = A specific human name falls within the whole vector space

$v_{job\ title}$ = A vector space for a specific job title

v_{name} = A vector space for a specific human name

To wrap up, after replacing all human names and job titles in each proxy statement report, we can continuously apply word2vec to those data to form word embeddings and calculate the possibility (i.e similarity) of a single name appears within a job title's context window.

Chapter 5
Analysis Process

After introducing all the techniques that we will apply to our data, we can leverage the named entity recognition and word2vec algorithms to identify who is the Chief Executive Officer in a certain company. In order to approach to our assumption, we will split the whole processes into three tasks:

1. **Clean Data**
   Including tokenizing, extracting human names, concatenating objects and replaced old objects (which are not concatenated) with the new objects (which are concatenated).
2. **Create Word Embeddings based on the data we cleaned**
3. **Relationship Analysis**
   Apply word2vec to the word embeddings and see the possibility (similarity) of a human name appears within a job title context window.

5.1 Clean Data

In order to increase the accuracy of our method, data cleaning is an essential part before we move on, especially proxy statement reports are usually in the unstructured formats. Considering the natural language processing tool (i.e. word2vec) we will use afterward, we divide data cleaning into two parts:

- Basic Cleaning: Remove html syntax, unrelated syntax like /n, /s and spaces
- Replace Full Names and Job Titles with Concatenated Names and Job Titles: The reasons have been illustrated in the previous section (i.e. Section 4).

Our data set includes 9 proxy statements from Tesla, Inc. There are roughly 250,000 words and 1,050 names within those proxy statements. After we clean the data, replace non-concatenated objects with concatenated objects, we have roughly 240,000 words as our corpus text.

	Output
Before	Tesla is committed to fair and competitive compensation for its employees. Moreover, Elon Musk, our Chief Executive Officer, has agreed to a compensation arrangement in the 2018 CEO Performance Award that is substantially tied to the appreciation of our market capitalization.
After (lower case and concatenated)	tesla is committed to fair and competitive compensation for its employees. moreover, elonmusk, our chiefexecutiveofficer, has agreed to a compensation arrangement in the 2018 ceo performance award that is substantially tied to the appreciation of our market capitalization.

Table 8: Before and After Cleaning the Data

5.2 Create Word Embeddings

From the previous step, we only do the fundamental cleaning jobs, which still cannot be used to analyze the objects relationship. Word2vec has two special attributes that:

- It can only take a single word
  → Was solved in the Section 5-1 as we replace human names and job titles with the concatenated/lower case names and job titles.
- It only takes in word lists

→ To be more specific, it only takes word lists from each sentences. Also, in order to measure the relationship between objects, each words do count as a vector space so that we cannot remove the stopping words like “the”, “at”, “a” etc.

	Output
Original Paragraph (After cleaning and replacing names and job titles)	tesla is committed to fair and competitive compensation for its employees. moreover, elonmusk, our chiefexecutiveofficer, has agreed to a compensation arrangement in the 2018 ceo performance award that is substantially tied to the appreciation of our market capitalization.
Normal word list (e.g. Bag of words algorithm) Have only one word list	List1: 'tesla', 'is', 'committed', 'to', 'fair', 'and', 'competitive', 'compensation', 'for', 'its', 'employees', 'moreover', 'elonmusk', 'our', 'chiefexecutiveofficer', 'has', 'agreed', 'to', 'a', 'compensation', 'arrangement', 'in', 'the', 'ceo', 'performance', 'award', 'that', 'is', 'substantially', 'tied', 'to', 'the', 'appreciation', 'of', 'our', 'market', 'capitalization'
Word lists for Word2vec Have two word lists	List1: 'tesla', 'is', 'committed', 'to', 'fair', 'and', 'competitive', 'compensation', 'for', 'its', 'employees' List2: 'moreover', 'elonmusk', 'our', 'chiefexecutiveofficer', 'has', 'agreed', 'to', 'a', 'compensation', 'arrangement', 'in', 'the', 'ceo', 'performance', 'award', 'that', 'is', 'substantially', 'tied', 'to', 'the', 'appreciation', 'of', 'our', 'market', 'capitalization'

Table 9: Word List Example

5.3 Relationship Analysis

After wrangling the data, we finally have approximately 7,000 word lists on hand (each list includes several words). Applying a word2vec model becomes straightforward with the help of "gensim" package, which contains a word2vec model.

When it comes to tuning the hyperparameters, there are often two ways:

A. Using the default values
B. Set hyperparameters based on your preference

Because the name frequency is often low within the proxy statement reports, we make up our mind to tune hyperparameters by ourselves, especially for three hyperparameters: the context window, the minimum frequency of a certain word and the dimension of a vector space (*all based on the gensim package document: https://radimrehurek.com/gensim/models/word2vec.html).

(a.) The context window (parameter: window)

This is exactly the window we named it in the previous section (i.e. Section 4.3). The definition from the "gensim" document is the maximum distance between the current and predicted word within a sentence.

According to the research from O Levy & Y Goldberg (2014), the larger context window may capture more topic-related information, whereas the smaller context window puts more focus on the word itself. Because our goal is to evaluate the relationship between a name and a job title, we are more curious about "words" instead of "topics". Therefore, we decide to use the context window as five, which means the maximum distance between the name and the job title within a sentence is five.

(b.) Frequency of word (parameter: min_count)

The function of this hyperparameter is similar to TF-IDF (Hans Peter Luhn (1957)), which the algorithm will ignores the word which its total frequency is lower than this number. The gensim package recommends us to set a number between 10 to 100. Because the name of Chief Executive Officer may not appear too frequently like 80 times within a report, to prevent from ignoring the important names (i.e. Chief Executive Officer's name), we decide to set this parameter as 10.

(c.) Dimension (parameter: size)

From the gensim package document, size means the dimensionality of the word vectors. In other machine learning cases, we regard the word vectors as the features. This gensim package recommends us to set a reasonable size like tens to hundreds. Usually, the larger amount of features turns out to have a more precise result with longer runtimes. Since we only have 240,000 words within 7,000 word lists, we decide to use 500 as our size.

After deciding the default values of our parameters within the word2vec model, we begin to build nine models which fit the data from year 2011 to year 2018.

As the models were built, we can simply have the "similarities" between a name and a job title by applying *model.wv.similarity* function to each model. For example, Figure 4 shows the name "elonmusk" (which was originally written in "Elon Musk") has 99% possibility that its vector space is close to the job title "chiefexecutiveofficer" (which was originally written in "Chief Executive Officer") in year 2018.

```
get_similarity(models[0], 'elonmusk', 'chiefexecutiveofficer')
```

```
0.9999347943937917
```

Figure 4: The Similarity Ratio between a Human Name and a Job Title

Furthermore, if we apply all the human names we collected and fit the models, we can clearly see that the name "elonmusk" has the highest similarity with the job title "chiefexecutiveofficer" in each year (See Table 10). These results show us the power of word2vec and how we can combine several natural language processing methods to find the name of the Chief Executive Officer.

Name	Year 2018
Elon Musk	0.9999
Deep Akahuja	0.9998
Kimbal Musk	0.9998
Jason Wheeler	0.9997
Doug Field	0.9997
Jon McNeill	0.9997
Ira Ehrenpreis	0.9996
Brad Buss	0.9995
Robyn Denholm	0.9995
Antonio Gracias	0.9995

Name	Year 2017
Elon Musk	0.9999
Jon McNeill	0.9996
Kimbal Musk	0.9996
Antonio Gracias	0.9996
Doug Field	0.9995
Draper Fisher Jurvetson	0.9995
Jeffrey Straubel	0.9995
Stephen Jurvetson	0.9995
Robyn Denholm	0.9993
Ira Ehrenpreis	0.9992

Name	Year 2016
Elon Musk	0.9998
Greg Reichow	0.9995
Jason Wheeler	0.9994
Doug Field	0.9993
Jeffrey Straubel	0.9990
Antonio Gracias	0.9987
Deep Akahuja	0.9985
Robyn Denholm	0.9985
Ira Ehrenpreis	0.9985
Kimbal Musk	0.9982

Table 10: Similarity Table Tesla in from Year 2016 to 2018

There are two advantages of this algorithm.

(a.) Can still identify the name of the Chief Executive Officer even it changes

The power of word2vec is, it takes each word as a vector and evaluates the distance between a name and a job title. That is to say, it does not take the name change or abbreviation into consideration. In this algorithm's perspective, all names are just vectors. Therefore, the methodology we used may not be impacted by the name modification because it only takes the position of that specific word into account. This is also the reason why our results remain the same even if we make a name to be concatenated and in a lower case.

Also, as the positions of names and job titles remain the same, even if the name changes after a marriage or has a different abbreviation, it can still identify who is the Chief Executive Officer.

(b.) See the turnover of the top management

We can also see the top management transition combining the time series model. As we may know that many companies have last for a long time, there may be somebody who took over the company. Previously, investors may know the top management transition by breaking news. However, by using the natural language processing, we can collect all similarity ratios of the human names and apply time series to the data. The result is like the Figure 5.

In Figure 5, the x-axis means the years and the y-axis means the similarities between the names we chose and the position title "chiefexecutiveofficer". The red line shows that Steven Jobs has taken over Apple Incorporation since 1999. During the time that Steven Jobs took over Apple, the future Chief Executive Officer - Timothy Cook also has a high possibility to be the Chief Executive Officer. In year 2011, the top management changed. Timothy Cook started to lead Apple, and we can clearly see the red line (which represents the similarities between "stevenjobs" and the title "chiefexecutiveofficer") went down as Steven Jobs withdrawn from his position.

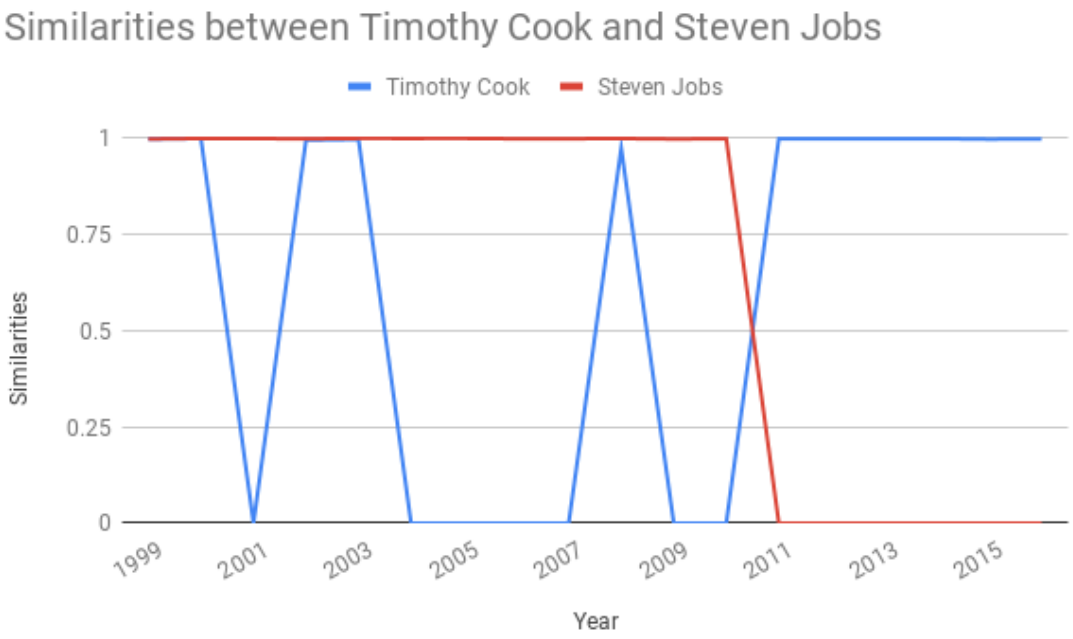

Figure 5: The Similarities between Different Names and the Job Title in Apple's Reports

To name it, this natural language processing technique can help us identify who has the highest similarity with the job title Chief Executive Officer regardless of name changes or time difference. It is because the similarity ratio shows ***Prob (human name | job title),*** which we can know the possibility of a human name falls within the context window of the job title.

Chapter 6
Conclusion

6.1 Summarizations

In this section, we hope to wrap up the two main concepts that this thesis shows:

1. How can we find the exact human names (full name) within a long text file.
2. How to identify who is the Chief Executive Officer from hundreds of human names

The aim of this thesis is to explore how can we combine several natural language processing techniques to approach a crucial question in the financial field. As we may know that the top management transition can have a huge impact on the financial market, investors may want to keep track of who is the Chief Executive Officer by year. After we extract hundreds of human names from the proxy statement reports, word2vec algorithm can help evaluate the relationship between a job title and a human name. Moreover, we can also see the turnover of the specific top management after applying time series to the similarity ratios.

6.2 Future Optimizations

Since we only apply our model to a small portion of data, the future challenges will be how can we optimize our model that it can broadly apply to financial reports from hundreds of companies. Because the scope of our research is relatively small, there may be several strategy to better optimize the scalability.

A. Use a more reliable data source

Most of the time, we use the proxy statement report (DEF 14A) to fit the model. It is because the proxy statement reports include variety of data like the board of directors, compensation tables, audits, finance reports and so on. This means we can use those data, which comes from different aspects to identify who is the Chief Executive Officer.

However, there are 158 types of reports in the U.S. Securities and Exchange Commission (SEC) Edgar database and we need to be cautious about the aspect that a specific report focusing on. Some of them, like the annual reports (10K) do not perform well because these reports are more focusing on the financial aspect. Since word2vec does heavily relies on the distance between a name and a job title, the output may be misleading if there are too many human names near a job title, which is a characteristic of the annual reports (10K).

The Table 11 and Figure 6 are good examples to show this bias. As we may know that Timothy Cook has been the Chief Executive Officer of Apple since 2011, our model mistakenly grants Luca Maestri the highest probability of being the Chief Executive Officer in year 2014 to 2016. It is because the annual reports have more financial contents involved so that Luca Maestri appears much more often as he was the Chief Financial Officer of Apple at that time.

Name	Year 2016	Name	Year 2015	Name	Year 2014
Luca Maestri	0.9988	Luca Maestri	0.9986	Luca Maestri	0.9993
Timothy Cook	0.9986	Timothy Cook	0.9985	Timothy Cook	0.9992
Polo Ralph Lauren	0	Polo Ralph Lauren	0	Polo Ralph Lauren	0
Mahendri Shah	0	Mahendri Shah	0	Mahendri Shah	0
James Buckley	0	James Buckley	0	James Buckley	0
Joseph Huber	0	Joseph Huber	0	Joseph Huber	0
Mark Kula	0	Mark Kula	0	Mark Kula	0
David Nagel	0	David Nagel	0	David Nagel	0
Avadis Tevanian	0	Avadis Tevanian	0	Avadis Tevanian	0
M Calderoni	0	M Calderoni	0	M Calderoni	0

Table 11: The similarity between names and the job title using the Apple annual report

Power of Attorney

KNOW ALL PERSONS BY THESE PRESENTS, that each person whose signature appears below constitutes and appoints Timothy D. Cook and Luca Maestri, jointly and severally, his or her attorneys-in-fact, each with the power of substitution, for him or her in any and all capacities, to sign any amendments to this Annual Report on Form 10-K, and to file the same, with exhibits thereto and other documents in connection therewith, with the Securities and Exchange Commission, hereby ratifying and confirming all that each of said attorneys-in-fact, or his substitute or substitutes, may do or cause to be done by virtue hereof.

Pursuant to the requirements of the Securities Exchange Act of 1934, this report has been signed below by the following persons on behalf of the Registrant and in the capacities and on the dates indicated:

Name	Title	Date
/s/ Timothy D. Cook TIMOTHY D. COOK	Chief Executive Officer and Director (Principal Executive Officer)	October 28, 2015
/s/ Luca Maestri LUCA MAESTRI	Senior Vice President, Chief Financial Officer (Principal Financial Officer)	October 28, 2015
/s/ Chris Kondo CHRIS KONDO	Senior Director of Corporate Accounting (Principal Accounting Officer)	October 28, 2015
/s/ Al Gore AL GORE	Director	October 28, 2015
/s/ Robert A. Iger ROBERT A. IGER	Director	October 28, 2015
/s/ Andrea Jung ANDREA JUNG	Director	October 28, 2015
/s/ Arthur D. Levinson ARTHUR D. LEVINSON	Director	October 28, 2015
/s/ Ronald D. Sugar RONALD D. SUGAR	Director	October 28, 2015
/s/ Susan L. Wagner SUSAN L. WAGNER	Director	October 28, 2015

Figure 6: Apple 2015 Annual Report - Power of Attorney Table

From SEC Database Edgar:

https://www.sec.gov/Archives/edgar/data/320193/000119312515356351/d17062d10k.htm

B. Optimize the named entity recognition algorithm

There is no perfect model to precisely evaluate the relationship between objects, and the natural language processing algorithms we used are no exception. Even though we can roughly extract 90% of human names, there are some names which were ignored by the algorithm. For example, if a part of name is followed by a salutation, like Mr. Cook, the algorithm will not identify it as a human name.

	Output
Original Input	Biographical information for Apple's executive officers, other than Mr. Cook, is listed below. Biographical information for Mr. Cook, who is both a director and an executive officer, can be found in the section entitled "Directors." In this section ("Directors, Corporate Governance, and Executive Officers—Executive Officers"), references to particular years refer to the calendar year.
Expected Output	Mr. Cook
Real Output	None

Table 12: Failed to Identify a Human Name

To sum up, this thesis provides an overview of how can we combine natural language processing algorithms like NLTK, Stanford Named Entity Recognition and word2vec to identify the relationship between a name and a job position. We applied those algorithms to proxy statement reports from Tesla and successfully identified the name of the Chief Executive Officer from year 2011 to 2018.

Reference

1) David Nadeau, Satoshi Sekine, “A survey of named entity recognition and classification”, *Lingvisticæ Investigationes*. Volume 30, Issue 1, pp. 3 –26, 2007
2) L.F. Rau, “Extracting company names from text”, *Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*. pp. 29 - 32, 1991
3) Michael Fleischman, Eduard Hovy, “Fine grained classification of named entities”, *Proceeding COLING '02 Proceedings of the 19th international conference on Computational linguistics*. Volume 1 pp. 1-7, 2001
4) Lee S., Lee G.G. (2005) “Heuristic Methods for Reducing Errors of Geographic Named Entities Learned by Bootstrapping”, In: Dale R., Wong KF., Su J., Kwong O.Y. (eds) Natural Language Processing – IJCNLP 2005. IJCNLP 2005. *Lecture Notes in Computer Science*, Volume 3651. Springer, Berlin, Heidelberg
5) Diana Maynard and Valentin Tablan and Cristian Ursu and Hamish Cunningham and Yorick Wilks, “Named Entity Recognition from Diverse Text Types”, *Submitted to Recent Advances in Natural Language Processing 2001 Conference*, Tzigov Chark, Bulgaria.
6) Zhu J., Uren V., Motta E. (2005) “ESpotter: Adaptive Named Entity Recognition for Web Browsing”, In: Althoff KD., Dengel A., Bergmann R., Nick M., Roth-Berghofer T. (eds) Professional Knowledge Management. WM 2005. *Lecture Notes in Computer Science*, Volume 3782. Springer, Berlin, Heidelberg
7) M Firth, PMY Fung, OM Rui, “Firm performance, governance structure, and top management turnover in a transitional economy”, *Journal of Management Studies*, Volume 43, Issue 6, pp. 1289-1330, 2006
8) F Wang, Y Xu, “What Determines Chinese Stock Returns?”, *Financial Analysts Journal*, Volume 60 Issue 6, pp. 65-77, 2004

9) Chapple, L. & Humphrey, J.E. J (2014), “Does board gender diversity have a financial impact? Evidence using stock portfolio performance”, *Journal of Business Ethics*, Volume 122, Issue 4, pp 709–723, 2014
10) D Nadeau, S Sekine (2007), “A survey of named entity recognition and classification“, *Lingvisticæ Investigationes*, Volume 30, Issue 1, pp. 3 –26
11) Luhn, Hans Peter (1957). "A Statistical Approach to Mechanized Encoding and Searching of Literary Information". *IBM Journal of Research and Development*. Volume: 1, Issue: 4, Oct. 1957, pp. 309–317.
12) KAREN SPARCK JONES, (1972) "A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL", *Journal of Documentation*, Volume 28 Issue: 1, pp.11-21
13) Mikolov, Tomas; et al. "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781
14) Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.
15) Omer Levy, Yoav Goldberg (2014), “Dependency-Based Word Embeddings”, *Computer Science Department Bar-Ilan University Ramat-Gan*, Israel
16) U.S. Securities and Exchange Commission (SEC) EDGAR Search Tools: https://www.sec.gov/edgar/searchedgar/companysearch.html
17) Nltk documentation: https://www.nltk.org/
18) Proxy statement: https://en.wikipedia.org/wiki/Proxy_statement
19) Gensim package document: https://radimrehurek.com/gensim/models/word2vec.html
20) Softmax function: https://en.wikipedia.org/wiki/Softmax_function
21) Word2vec (skip-gram model): PART 1 - Intuition: https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b