Kabore Kader Monhamady

DEVELOPING MACHINE KEARNING METHODS FOR BUSINESS INTELLIGENCE

AGU 2018

Developing Machine Learning Methods for Business Intelligence

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF NATURAL SCIENCES OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER

By
Kabore Kader Monhamady
November 2018

Developing Machine Learning Methods for Business Intelligence

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF NATURAL SCIENCES OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER

By
Kabore Kader Monhamady
November 2018

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Kabore Kader Monhamady

REGULATORY COMPLIANCE

M.Sc. thesis titled "Developing Machine Learning Methods for Business Intelligence" has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Kabore Kader Monhamady	Dr. Zafer Aydın

Head of the Electrical and Computer Engineering Program

Prof. Dr. Vehbi Çağrı GÜNGÖR

ACCEPTANCE AND APPROVAL

M.Sc. thesis titled Developing Machine Learning Methods for Business Intelligence and prepared by Kabore Kader Monhamady has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

25 /12 / 2018

(Thesis Defense Exam Date)

JURY:

Dr. Zafer Aydın

Dr. Bekir Hakan Aksebzeci

Dr. Mete Çelik

APPROVAL:

The acceptance of this M.Sc. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / …….. and numbered ……..……. .

…….. /…….. / ……..

(Date)

Graduate School Dean
Name-Surname, Signature

ABSTRACT

Developing Machine Learning Methods for Business Intelligence

Kader Monhamady KABORE

M.Sc. in Electrical and Computer Engineering Department

Supervisor: Dr. Zafer Aydın

November-2018

Detection of key attributes in text is an area of research, which attracts attention due to the increase of data and the availability of massive documents. Key attributes serve as metadata for documents and the discovery of accurate characteristics allows to capture significant pieces of information from a lengthy text. They allow faster and efficient information retrieval on the web domain with an ever increasing number of websites. In this thesis, a novel two-stage machine learning method is developed to identify the company name from web page text. The problem is reduced to a classification task at the token (i.e. word) level followed by a post-processing phase for predicting the company name. Features are extracted using natural language processing techniques and by observing patterns present in textual data to reflect the properties and significance of the words in context. Derived features are sent as input to classification algorithms such as naive Bayes, decision tree, and random forest. In addition to the token-based classifier, a rule-based method is designed that also considers tokens from domain as well as page title and ranks tokens by computing similarity metrics. The results demonstrate high precision from the machine learning model along with high undefined cases whereas the rule-based approach obtained high accuracy with precision inferior to the token-based model. When the two classification strategies are combined into a two-stage classifier, high accuracy and precision scores are obtained.

Keywords: Named Entity Recognition, Company Name Detection, Natural Language Processing, Web Mining, Feature Extraction, Machine Learning

ÖZET

İş Zekası İçin Makine Öğrenmesi Yöntemlerinin Geliştirilmesi

Kader Monhamady KABORE

Elektrik ve Bilgisayar Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Dr. Zafer Aydın

November-2018

Anahtar özelliklerin tespiti, verilerin artması ve büyük belgelerin daha hızlı ve kolay erişilebilir olmasından dolayı giderek ilgi duyulan bir araştırma alanıdır. Anahtar özellik, belgeler için meta veri görevi görür ve doğru özelliklerin keşfi sayesinde, uzun metinlerden önemli bilgi parçalarının yakalanmasını sağlar. Anahtar özellikler, internet alanında giderek artan web sitelerinden daha hızlı ve verimli bilgi keşfetme imkanı sağlayabilir. Bu tezde, verilen bir web sayfası metninden şirket ismini otomatik olarak tespit eden iki aşamalı yeni bir makine öğrenmesi yöntemi geliştirilmiştir. İlk aşamada verilen bir kelimenin şirket ismi olup olmadığını tahmin eden bir sınıflandırma yöntemi geliştirilmiştir. Yöntemin kullandığı öznitelikler doğal dil işleme teknikleri ile ve metinsel verilerdeki örüntülerin incelenmesi sonucu kelimelerin özelliklerini ve içeriğe ilişkin anlamını yansıtacak şekilde çıkarılmıştır. Bu öznitelikler daha sonra naive Bayes, karar ağacı ve rastgele orman gibi sınıflandırma yöntemlerine girdi parametresi olarak aktarılmaktadır. İkinci aşama içinse kural tabanlı bir sınıflandırma yöntemi geliştirilmiştir. Bu yöntem alan ve başlıktaki kelimelerini de tarayarak simge benzerlik ölçütleri ile şirket ismi olmaya aday olan kelimeleri sıralamakta ve en yüksek skorlu kelimeleri şirket ismi olarak tahmin etmektedir. Yapılan deneyler sonucunda birinci aşamadaki sınıflandırıcı ile yüksek hassasiyet oranı elde edilirken özellike zor olan bazı metinlerdeki şirket isimlerinin tanımsız kategorisine atandığı gözlenmiştir. Diğer taraftan kural tabanlı sınıflandırma yöntemi ile yüksek doğruluk oranı elde edilmiştir ancak bu yöntemin hassaslık oranı birinci aşamadaki yöntemden daha düşüktür. İki sınıflandırıcının birleştirilmesi sonucu elde edilen iki aşamalı sınıflandırma yöntemi ile hem genel doğruluk oranı hem de hassaslık oranı yüksek olarak elde edilmiştir.

Keywords: Adlandırılmış Nesne Tanıma, Şirket Adı Tespiti, Doğal Dil İşleme, Web Madenciliği, Öznitelik Çıkarma, Makine Öğrenmesi

Acknowledgements

I would like to express my sincere appreciation to my advisor Dr. Zafer AYDIN for his excellent support, assistance and mostly for the great patience he endured to assist me throughout my master degree. Patience is the highest degree a teacher offers to his student.

I would like to thank my friends and my family members particularly my Mother Mariam ZABRE for her in infinite love.

This work is part of a research on the field of web information retrieval and classification for CREDE ANALYTICS.

Table of Contents

List of Figures

List of Tables

This thesis is dedicated to all my beloved Mother

Chapter 1

Introduction

In the definition of the big data [1] the fifth quality refers to the value. In fact, data is ubiquitous and can be observed at every corner where technology resides, however the gathering of relevant data become a challenging problem. The acquiring of relevant data demands a step of transformation and considerable work to obtain essential information from a massive amount of data. These processes are frequent and can be seen in areas where textual content is at the core of the analysis. Text occupies a large percentage in the entire quantity of the concept called big data. Modern applications such as news, social networking sites and emails offer the possibility to exchange textual data and contribute to increasing the volume of text in database and warehouses. Along with this increase of content, rises the need for techniques and tools with the capability of converting the content into simple, more understandable and more utilizable data. The challenge is presented as the inquiry of the best applicable solution to extract knowledge from data. Various alternatives are being studied in order to offer the best results for information extraction.

In this thesis, we investigate a solution for the detection of the company name within a web page. A normal website has one or more pages which the purpose of the website. Each page holds some content where the company name of the website can be found. The company name is the main entity associated with the domain of the website. We elaborated steps and proposed appropriate methods for the accomplishment of the company name detection. The works main focus was on text processing. The study relied on the existing work in the field of text processing and the contribution of innovation techniques derive along the analysis by means of deep examination of the domain and the data provided.

The following sections include some definition for the field of text processing and preceded with the literature review and contributions of this thesis.

1.1. Text processing

In this new era, computer networks reached a pic in advancement. Interaction and communication have become easier, faster and possible from every corner of the world and lead to the internet. The internet is a massive connection of computers and machine in a global network. The internet drew to the center of all activities. Computers became the backbone of all interaction and information. Machine-readable documents become available. Computers are programmed to serve and respond to the user's request. Computers offer an interface to end user to interact. These interactions occur generally in a human understandable way through voice, written language, signs. The interactions are collected and stored in a text format to allow human readability.

As a result, this the bulk of business-relevant information originate from in raw format, unstructured form and primarily text. In fact, a recent analysis of IBM revealed that 80% [2] of the current information of companies are contained in text format. This data need to be transformed from human understandable format to computer acceptable and process able format. Text processing envelops all these fields which seek to transform raw text into meaningful and more informative data.

1.1.1. Text Mining

Text mining is the branch of knowledge which deals with text as it tries to recover relevant knowledge from textual data. It emerged in the 80s along with the rise of business intelligence and the need of converting business related transactions into meaningful information [3]. It is part of the Artificial intelligence stack and put the concerns on data represented in text format. Text mining is a variation to the field called data mining which aims at finding a relevant pattern from large databases [4].

Text mining is the process of discovery and extraction of essential information from unstructured textual data. It is an interdisciplinary field and draws on statistics, machine learning and computational linguistics. The role of the text analysis is to disclose any form of the pattern that could be contained in a dataset made of text. Text mining incorporates several applications that hold a tremendous result in the domain of technology.

1.1.2. Information Retrieval

Information retrieval is the finding of proper documents that meet the query assigned. This part of computer technology started to get attention after the growth of large collections on the world wide web. The growth of documents around the web renders old techniques of documents retrieval ineffective. A new science was necessitated to produce efficient methods and techniques for the purpose of retrieval of appropriate content and also in a fast manner [5]. Information retrieval has the key goal to go through the whole content and fetch the best documents holding relevance to the request. This goal is achieved by means of statistical measures and methods applied for the automatic processing on textual data given a connection to the inquiry text. Information retrieval is well known for the success the field impacted in the world wide web. One of the best applications of information retrieval is search engines. Search engines play an important role on the internet and hold the task of responding to user search queries and bring back the sought pages that best interest end users.

The data collected on the web is usually immense and need to be stored in an efficient manner. Search engine implements a powerful mechanism for accomplishing the process of storing. The data is sliced into a small unit of words and hashed [6]. Each unit is then stored at the appropriate location given the hashed of the token to allow faster retrieval. The last component of a search engine is the search part. The search portal is the section visible to end users. The remaining components are hidden and the user interacts to all system through the search function. Search is the action of retrieving desired information given a query. The user places a query by accessing the interface of the search engine and establishing a request. The request is accessed by the search engine and regarding the data indexed fetch the best documents that meet the request of the user. In order to show the relevance of each document and score is associated with each page in accordance with the importance, the document holds to fit the request. These pages are sent back to a display where pages are ranked according to the priority of the scores [6 see chapter 5]. Users can access search easily and locate proper document quickly and adequately. Search engines are at the center of the internet. It manages transaction around the world wide web.

1.1.3. Information Extraction

Although the data can massive and large, the relevance can be of a tiny value. The value of the documents is determined by the significance of the information extractable. Large collections usually have to undergo some preliminary steps to get rid of the useless information and expose real knowledge enclosed. Information extraction is a mechanism which takes place along with text processing related tasks. It consists of procedures which seek to highlight relationships between entities on one hand and on the other hand discard parts of the data with low meaning. It decomposes into processing steps including sentence segmentation, tokenization, and detection of entities [7]. The main task is to identify significant sections and assign related attributes to it. As an example, we can illustrate the function of extraction place names from news documents or capture username from blogs posts. These applications are frequent in the modern era over flooded with text documents.

1.1.4. Natural Language Processing

Natural language processing referred to its origins to the previous decade. Natural Language Processing research notions appeared in the years of the 1940s [8]. In fact, the first results of computer achievement were observed in Natural language processing. The first applications were machines translator conceived to break the enemy Codes from the world war 2. The research went ahead at various college and institutions, however, the actual focus and initiated after the work presented by Noam Chomsky. Noam Chomsky made a publication of syntactic structure which layered a foundation and significant approach to the field. Earlier work on Natural language processing focused on abstracting the language as a simple model and represent the track. Noam Chomsky research targeted at generating linguistics patterns and rules well known as grammar [9]. His work introduced a new way of generating and representing the syntactic pattern of a language. This redefined the main role of the field.

In fact, natural language processing is a particular track of artificial intelligence dedicated to human language. Natural language processing seeks to render computer more understandable of natural language utilized by humans. In the current state, computers receive a request through a programming paradigm. Human communicates to

machines by a set of rules encoded into programming languages. Some examples of programming languages are Java, C++, python etc.. Natural language processing aims at facilitating the interaction between machine and human with a direct and more convenient access. It covers discipline involving computer science and general linguistics as well. Basically, this field is classified into two components. The first component is natural language understanding which attempts to enable computers to understand directly human language without a programming interface. It deals with the phonology which the way the words sounds, the morphology which refers to the way the words are shaped, the syntactic structures, semantics, and pragmatics of the language [8]. The figure below shows the broad field of natural language processing. And the second component is the natural generation which puts focused on means to produce a language from machines similar to human language. Natural language understanding tries to improve human-computer interaction from the existing interfaces to an intelligent and user-friendly manner.

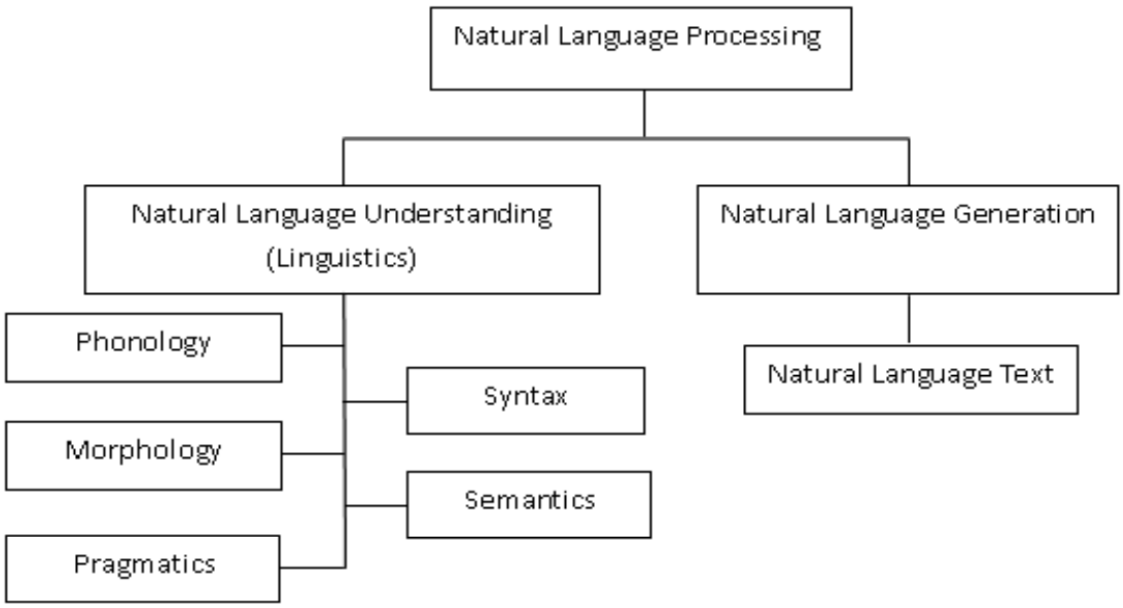


Figure 1.14.1: Broad Classification of Natural Language Processing [8]

1.2. Entity Detection

1.2.1. Rule Based Approach

Numerous are the problems involving the extraction of a special entity or words within a large amount of text content. Over the years' various solutions and techniques have been proposed to deal with the detection and generation of special entities. The early methods were straight forwards and proposed rules based methods for the extraction of researched words. Before the rise of machine learning, regular expressions were the best way to illustrate a pattern in text processing. The expertise of linguists was utilized to build grammars and these grammars were used to restrict unwanted tokens and capture desired tokens [10]. The results were time demanding tasks and were not robust news introduce textual data, though the results were by some means satisfying at that early time and the best approached presented results near to machine learning approach. Téllez-Valero, Alberto et al. [11] demonstrated this approach by building an extractor which detects information related to disaster from news articles by means of recognizing words around the relevant information. Results were also observed in e-commerce applications where similar techniques were applied to filters products attributes [12]. Azimjonov, Jahongir et al. [13] utilized this paradigm in academic researches. They proposed a rules-based extractor which detects metadata from academic articles. They produced an accuracy of 91.21% for the metadata extraction in the title and 92.53% for the keywords and the index terms.

1.2.2. Machine Learning Approach

The rise of statistical methods and the advancement offered more opportunities for specific entity detection. The ease and the simplicity exposed more problem opened for solutions through a machine learning approach. One of the simplest assignment was redundancy removal. Redundancy removal compromises of identifying redundant information from documents and text to facilitate further analysis of applications such as questions answering and summarization [14]. Another attempted assignment is keywords extractions. This assignment is approached through two procedures.

The first is an identification procedure. It concerns processing the text and detects the best possible candidates to be referred to as keywords for the context. The text is

processed entirely using all the words it contains.By means of features including the appearance count of the word and the location in the text a prediction is made determining the probability of the word depicting as a keyword. Keywords extraction is challenging and the results can be deficient. Tanya Gupta [15] in her work tried to apply the best algorithm for the tasks of keywords extractions and achieved an F1 score of 24.63% and an F2 score of 21.19 %. Related works were undertaken to figure the 5 w's of a textual document. It follows a philosophy which states that a document can narrow down to 5 essential questions such as the who, what, when, where, and why [16]. The answers to these five questions are the summary of the textual document. The procedure is usually called focus name extraction or context based title extraction.

A second alternative to the identification approach is a generative approach. In this fashion, the sought entity is not directly located within the text content. The textual data is analyzed and a suggestion is proposed to stand as the entity requested. The purpose is to make use of the knowledge of the content and generate a key phrase with the capability of representing the content. [17] proposed an automatic key phrase generation which encodes the context into the title-guided representation. They applied deep learning methods and surpassed the state of the in key phrase prediction.

1.2.3. Web Mining

Web mining is part of the general field of text mining. The field is recent, however, receive a lot of attention with the rise of the world wide web and the huge amount of data available on the internet. Web mining deals with three main aspects: The content of the web, the structure of the web and the usage of the web [18]. The main goal is to gain insight and knowledge from the data around the web. The Internet has become nowadays the mainstream of information for users nonetheless this information often appeared unstructured and massive. Mining this data into relevant information is the primary objective of text mining. Applications of text mining draw on system improvement, web personalization and business intelligence where the elements of the page are used to make an excellent prediction and drive more sales.

Information extraction from HTML was one of the early challenges in the field of web mining [19, 20]. The possibility of acquired key data from the page was a reality and several options have been investigated [19] to show the importance of value contained in

HTML tags. They demonstrated a rules learning to extract product information from pages. Xue, Yewei et al [21], in their research, used the layout of the pages to extract the title of the page. Web pages have a different layout and the way the content is delivered is deterministic of the position of the title. The document model of the page is parsed and specific tags are located to extract the title of the page. Gali, Najlah and Pasi Fränti [22] also approached the problem in a similar fashion and added a processing step where the term frequency and the document inverse frequency is computed to assign a higher value to frequents appearing token. A patent was suggested for the extraction with a contextual implication [23]. The frequencies of the words extracted from the title, the body, and the URL were calculated and the contextual title by comparing the frequencies. The results were applied in internet navigation browsers to suggest a title to tabs when there are multiple tabs in a window browser. The intent is to help users navigate easily through tabs using the title suggested. Packages extraction for keys entity on the web and social site are still under development and at an early stage [24]. Nonetheless, the field is in rising state with the business demands in the field.

1.3. Contributions of the Thesis

The web is a large resource of information and is a center of research due to the frequently increase of data rendered on daily basis. Web search engines are currently the third party in charge for extraction of the information on the web and the delivery of the retrieved information to the intended request. Although search engines have proposed acceptable solution to the extraction of page contents in a general ranking approach, an adequate solution to the extraction of specific attribute for single pages are still under a research phased. Websites are constructed to host a vast number of pages and further, pages are being designed to hold large descriptive contents. Thus, methods for detection of explicit attribute in a page is an attractive solution on top of existing search methods.

In the literature, solutions were proposed in general documents including books and newspapers where the objective aimed at determining the best keywords in the documents [13].

In this thesis, a solution is proposed to the attribute extraction on the web page level. The attribute chosen is the company name represented in the webpage. A page usually contains the company name of the website and drives the company name as an

important attribute for the webpage. This attribute can be used for better indexing for pages and an excellent search criterion for the content.

The contributions of this work can be divided into two levels. The first level is on the data layer. Previous works, in the field, made use of the HTML content in their analysis which allowed to extraction according to the HTML tags and ignore the text content of the page. In this thesis, HTML tags are not included in the data and the feature are extracted based on the textual content. This shows the importance of the text and the relevance it contributes in text mining and web mining.

The second level concerns the methods put in practice for the solution. On one side, two based level prediction are suggested in order to train and evaluate the model to detect the company name in the page. A token based to reduce the problem to a classification task where the features of delicately selected through various techniques and combined with features of existing natural language processing project. The training is achieved with machine learning algorithms and parameters are optimized to provide the best accuracy. A global level to allow a single prediction for the webpage where the token prediction and reprocessed to point out the perfect candidate for the page. On the other side, rules based predictions are highlighted and a pipeline prediction queue is explored by combining the best results of the ruled based prediction along with the outcome of the training analysis. The pipeline increases the confidence of the prediction and make the model robust to errors predictions.

Chapter 2

Data and Methods

2.1. Dataset

The dataset was generated by CREDE analytics. They performed the crawling of web pages from multiples domain in order to construct the dataset. The crawler tried to gather content from the first page or well known as the about page where the website company presents detail and information of the company profile and their achievement. The original data set contain up to 100.00 rows of the crawl process. They were categorized into two separate sheets of excel contain. The result of the annotation produced and allow to retrieve 1000 clean rows having appropriate textual data for the analysis.

2.1.1. Description

The dataset in the analysis is the result of a process of crawling web pages. It has some differences to standard datasets. Regular datasets are shaped in rows and columns where each column describes a value related to the record. This dataset is assembled from parse outputs of a web crawler. The crawled data is not from a specific field or specific HTML tags. The whole content of the pages is collected and is parsed as text. The crawler was instructed with parameters to extract specific content from the websites.

The specific fields retrieved are as follows: (1) IntIntroductoryTextID field shows a simple integer value of the introductory text. This value is correlated to the introductory text fetched by the crawler. (2) StrIntroductoryTitle field depicts a short sentence from the website as the welcome text. It is taken from the section from the web page to illustrate the title of the page. A website is usually composed of several web pages and each has a title to identify the specific pages. (3) StrIntroductoryText field represents the real content of the webpage. It is the direct outcome of the parser of the mark-up text. The content varies from simple words to long paragraphs of text holding the content of the webpage

encoding in both the body and the heading HTML tags. This field is the main attraction of this research by the value it holds. The value of this field is used text to build models in order to detect the company names it contains. (4) IntLegalInstitutionID field holds integer values which stand for the institution of the particular website. The crawler generates value to represent the legal institution of the web pages. (5) IntDataSourceID field represents the actual source of the retrieved website. The values are in a number format and are unique for each website gathered by the web crawler. (6) dtCreate field shows the date of creation of the webpage. The date since the website was available to users on the web. (7) strUrl field is the domain URL for the particular page processed by the crawler. It denotes the path of the content retrieved especially when the page is not the main page of the website. (8) IntLanguageID field holds finite number values and illustrates the language of the web page. Each value is representative for a distinct language of the webpage.

Among these columns, only two were exploited in the experiment: The column containing the raw text of the webpage (StrIntroductoryText) and the column containing the URL path (StrUrl) of the webpage.

2.1.2. Annotation

2.1.2.1. Methods

Labeling or data annotation is the task of associating the right labeled to the right class. Data already labeled is expensive. Data generated from databases are mostly labeled with a field categorizing each record. However, data acquired by means of crawling or scraping are often gathered in a raw format without labels that characterize each class. Therefore, beyond the generation of a dataset, each record needs to be assigned the appropriate category as a task called labeling.

The first alternative is a manual annotation of the data. The data is analyzed deeply and carefully at each level and the appropriate class is assigned to the records. Manually labeling the dataset is an exhaustive work and a time-consuming task. As a result, the size of the data can render this process impossible. The data gathered on the web can be tremendous and human effort can be deficient for this tasks.

An alternative choice is semi-supervised labeling. Semi-supervised labelling is the processing of annotating the dataset including both methods of manual labelling and supervised prediction of the model [26]. Semi-supervised labeling is divided into two phase. The first stage requests to label manually a small portion of the dataset and train a model with the section labelled [27]. The second stage involves labelling the remaining of the data using the trained model parameters. It results in an entirely labeled dataset with a portion manually annotated and the left of the data automatically annotated. In the case where the labels are not well-determined semi-supervised can be an ideal solution to produce the fastest outcomes [27 see chapter 3].

2.1.2.2. Labeling Issues

For this research, a supervised labeling was impracticable due to some reasons as follows. Some web pages were fetched by the crawler although the site was not functional. Some web pages were just content of programming code or warning message about the current state of the website. The websites retrieved were under construction for their new site or completely shut down. As a result, the text retrieved for these web pages is not significant to derive the company name.

Some websites restricted the access of their web pages to the crawlers. Crawling can slow down a web system and eventually cause harm to the site. So for security and performance reasons some website forbid the crawling of their site particularly to crawlers which do not abide by their politeness policy. Consequently, some contents were just empty of text and without any relevance for the analysis.

The central point of the research is to determine the company name from the webpage which implies the company name contained on the page. Some web pages retrieved were not the main page of the site and do not include the company name within the page content. The lack of a company name in the content makes the page ineligible for the experiment, especially for the training phase.

2.1.2.3. Application

As a result of the above issues, the dataset was manually annotated. A subset was randomly selected from the original dataset. Random samples were assembled to go through the labeling process. Each page within these subsamples was examined to check

whether the content is valid for the experiment and contains a predictable company name. The pages that do not comply with the validity process were discarded to have a cleaner data set. From the selected pages, each page retained data is then assigned with a word or a group of words that portrays the company name. The validity of a webpage consisted of checkpoints and criteria to confirm the state of the page content. The table 2.1.2.3.1 shows the attribute put into consideration for the selection of pages.

Criteria	Explanation
Content	Check whether the content is empty or contains distorted words like programming codes.
Content + URL	Check whether the content is really originated from the URL crawled.
Tokenizable	Check whether the content is clear and the content can be tokenized.
The language of the webpage	Check if the textual content of the page is written in the Latin alphabet.
Company Name	Check if the content is a page with the company name in it

Table 2.1.2.3.1: Data selection criteria

2.1.2.4. Annotation Ambiguity

Ambiguity is one of the most recognized issues in natural language processing [28]. This issue highlights the complexity of text in general and human languages in particularly due to fact that communication is shared between two entities which must share the same processing manner in order to understand one another. Indeed ambiguity touches most of the subfields of natural language processing [29] and has a great impact on the advancement of the research domain. Some solutions were initiated to in some area

line name entity recognition [30] but the results are still far from perfection. Ambiguity touches the company name detection as well. Ambiguity regarding the company name is the main issue in the annotation tasks. The company name is the central title which depicts the website with all pages incorporated. It is a business name discovered by the founder of the company and assigned to the website. It reflects the business name even outside the site representation.

First, the company name can be n-gram tokens. It can be unigram words of the size 1 word, bigram words of the size of two, trigram words of the size of 3, quadrigram of the size of 4 and so on...This makes the company name ambiguous such that the company name is indistinguishable from the surrounded tokens. The bigram representing the company name can be confused with a trigram or a quadrigram group of words. (e.g. Grand Star Hotel Bosphorus).

In addition, the company name can take different shapes at different places of the page content. In cases were observed where the company name is composed of more than a single it occurs circumstances where the tokens are combined into a single string at specific places in the webpage. This makes the company name take more than one form throughout the web page.

Moreover, abbreviations and short forms can take precedence over the original and expanded term of the company name. Some companies despite the fact that the full and longer form of the name stands for the company name, the company is well and better recognized in the abbreviation. As a consequence, the names can be multiples consisted of abbreviations and full expanded form (e.g. National Space Society | NSS).

Furthermore, some universal words are often associated with the companies' name and are not clearly differentiable from the actual company name. These universal words vary from sectors of activity and languages as well (e.g. Mandarin Oriental Hotel Group).

In the experiment, the ambiguity is resolved with the redundancy of the possible occurrences of the company name [14]. The label was extended to all words and group of words with similarities and potential to denote the company name. Acknowledging multiple shapes and aspects of the desired company name allows to captures diverse possibilities of the company name on the page.

2.2. Preprocessing and Feature Extraction

The data was generated in raw format. Unlike regular data mining analysis, the feature were not incorporated and given at the starting point. In order to approach the data for an excellent analysis the data is required to pass through pre-processing pipeline and a feature engineering task.

2.2.1. Tokenization

Tokenizing is the first phase towards processing content generated from text. Text generally has a different structure and is difficult to feed into an algorithm or task. Text requires an earlier step called pre-processing where the content needs to prepare to meet a standard format for computers to access the content data and extract parameters for algorithms. The main purpose of the pre-processing is to standardize the text into units having meaningful information. Tokenizing is the main component of in the section of pre-processing. It aims at converting the text unstructured and organized in the paragraph into token easy accessible by computers. It first reduces the content into sentences by detecting relevant pattern such as commas, full stop, characters cases etc.. These sentences are then inserted into a second pipeline where words are derived from the tokens. A token represents a unit, a piece from the original text. Regarding the format of the sentence individual word are split as an array of tokens. In this procedure, the original content only readable by a human is transformed into a format readable and manageable by computers. Although the text is not comprehensible by a computer, tokenizing provide an efficient approach to convert content into token units accessible for information extraction and understanding of the message of the text.

In this research, we went beyond the regular word base tokenizing. The simple tokenizer produces token consisting of single words of characters. However, the company name can be a collection of more than an individual token. We added a second layer of tokenizing to apprehend n-grams company names. The second tokenizer combines tokens from the first pass having title case. Company name composed of multiple words are generally in form of title case. Thus, consecutive words are merged into a token if their first character is upper case to facilitate the detection of features of n-grams. Similar alternatives were used in tasks such as Name Entity recognition where a possible class

was attributed to each token such as Begin, Inside, Outside, Last token. This suggestion of predefined classes is consistent to boost and produce improve accuracy [31].

2.2.2. Features Extraction

Regular data mining projects involve dataset where the inputs feature X are provided in order to deduct the output Y. In text mining and, particularly, in the text language related analysis the features are produced through a generative approach. The features are not provided along with the textual content but rather are derived by means of feature extraction process. The expertise of the domain and prior knowledge of the data is required by the feature extraction. A feature function is assigned to extract properties for each individual token contained in the textual data. This function is an integral part of a text mining project and deciding for the perfect features is the essential part of the analysis.

Features are depicted in various indicators. An indicator takes a value 1 or 0 if to indicate the presence or the absence of the pattern desired. A count can be also an indicator and illustrate the number of observations of the pattern. Feature extraction in this analysis necessitated to carefully explore the textual data of web pages and highlight common similarities on one side and on the other side utilized well-used features in the domain of text mining and natural language processing. The features are mainly divided into two categories. A local category which takes care of deriving features pertaining to the word itself and a global category which extracts features according to the context presented. In fact, an early solution focused on the local features but context revealed to improve and give greater results [31, 23].

2.2.2.1. Local Features

Local features extraction deals with deriving features patterning to a single token regardless of the context. The main focus is syntactic and the phonetic qualities of the word. It gives a good perception of the word itself and valuable insight into the shape of a company name.

2.2.2.1.1. Properties of word

These features are extracted based on the properties of the token. It tries to captures important pattern regarding the token properties. The table 2.2.2.1.1 shows the list of features extracted from the word properties.

Feature Name	Attribute
Prefix-1	The first prefix character of the token
Prefix-2	The second prefix character of the token
Prefix-3	The third prefix character of the token
Suffix-1	The first suffix character of the token
Suffix-2	The second suffix character of the token
Suffix-3	The third suffix character of the token
Lower	The lower case of the token
Stemmed	The stem of the token
Word-len	The length of the characters

Table 2.2.2.1.1.1Features extracted from the word properties

2.2.2.1.2. Word shape

These features are extracted to show the relevance of the shape. The shape can be used as a feature and has been used in natural language-related tasks [33]. The table below represents the features list of the word shape.

Feature Name	Attribute
Is_title	Check if the token start with an uppercase letter
Is_lower	Check if the token is all lowercase
Is_upper	Check if the token is all uppercase
Is_digit	Check if the token is a number
Is_camelcase	Check if the token is in camel case shape
Is_abbv	Check if the token is an abbreviation
Has_hyphen	Check if the token contains a hyphen separator
Has_dot	Check if the token contains a dot.

Table 2.2.2.1.2.1. Features extracted from the word shape

2.2.2.1.3. Word Type Features

The type of the word can be represented as features. There are existing lists of prefixes and suffixes which allows identifying the type of the words. These types are extracted and proposed as features in the experiment. The word type features are illustrated in the table below.

Feature Name	Attribute
Person_prefix	Contains or starts with the person name prefix(i.e. Mrs)
Person_suffix	Contains or ends with the person name prefix(i.e. jr)
Organization_suffix	Contains an organization suffix (i.e. corporation)
Nationality_In	Contains a country name
Location_In	Contains a location name(i.e. province name)
Numeric_in	Contain numeric value

Table 2.2.2.1.3.1. Features extracted from the word type

2.2.2.2. Global Features

The global features seek to represent meaningful features of the token with reference to the context. The extraction tries to bring a relation of the focus word and the remaining tokens in the text. These features show some relevancy of the token in the web page web. The global features are further subdivided into subcategories.

2.2.2.2.1. Cascading Features

Some features are derived in a cascading style. The output prediction of one class is reinserted to produce a second feature useful to serve as a feature input for the main

prediction. This fashion is efficient at producing valuable inputs with the abilities to represent the token in the sentence.

2.2.2.2.1.1. Part of Speech Tagging

Part of speech tag is an excellent feature included in the majority of text mining processes. Human language generally complies with a grammar which the way words are arranged together. Each word stands and represents a specific tag in a sentence. The main goal of part of speech tagging is to detect and assign the appropriate tag to each token within the sentence. Early works in the field approached the problem with rule-based methods [34]. However, the recent growth in the machine learning and deep learning [35] contributed to the field and delivered results close to human level accuracy.

In our analysis posterior the tokenization process, the tokens are inserted into a tag detector model where a tag is associated with each token. The tag generated is included as a feature for the token and the neighbour in the pipeline. In the English language, the part of speech follows a certain standard. The part of speech is in a number of 35 from the NLTK framework [36, 37].

2.2.2.2.1.2. Name Entity Recognition

Name entity recognition is an important task in the field of natural language processing. The process consists of identifying words entities in a sentence. Entities are often called proper names in human language. These entities are unique nouns or group of nouns which denote person names, location names and so on. The task involves two main phases: Name entity identification where the unique nouns denoting special entities are correctly retrieved from the sentences and Name entity classification where the retrieved entities are classified into predefined categories of entities. Name entity recognition is carried out by means of machine learning prediction with algorithms like a Conditional Random field, Hidden model [38] and recently deep learning method have surpassed the state of the art [39].

2.2.2.2.1.3. Semantic role labeling

Semantic role labeling is the task of detecting related part from sentences. In general, language is structured with interconnected parts to give a meaning. Each word is interconnected to the remains word of the sentence. Each word gives a meaning in the

context according to the dependency it has in the sentence. Each part plays a role in the thematic and justifies its position in the sentence. Common semantic roles proposed are agent, patient, instrument, beneficiary, source [40]. Agent or well-known as a subject in most languages is the entity of the action. The entity goes through the process of the verb. The patient is the direct object highlighted in the sentence. The patient is the impacted entity of the action. Instrument the entity used in the action by the subject. It is the means of the subject. A beneficiary is an entity that gets profit from the action. It is the end and the entity that gains from the action. The source is the entity that starts from or the object from the action. It is the origin of the action. These components are interlinked to form a sentence. Semantic role labeling is important regarding the contribution the task add the field of natural language processing. Direct applications of semantic role labeling are question answering and information extraction.

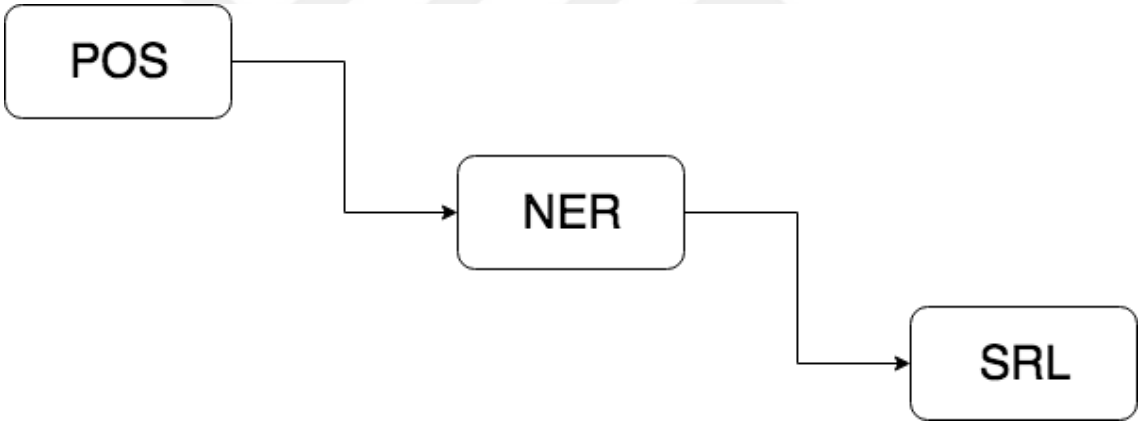


Figure 2.2.2.2.1.3.1 Overview of cascading features

2.2.2.2.2. Web Content Features

Beyond the standard pattern of the token as a part of a paragraph and sentences, extra feature is required to emphasize the value of a company name. A web page is not well structured like a paragraph, nevertheless, it contains hints on the page around the company name and gives a good intuition to make an excellent prediction. Some keywords, special characters, and some similarities are really determinative to indicate the presence of a company name among the closest words. Company names are generally

followed with some short abbreviation within a certain window of tokens. The results of the manual feature extraction are summarized in the table below.

Features Name	Explanation
Window	Features of the surrounding tokens on the left and on the right side
Trigger	Check if the surrounding tokens are contained in the trigger list of words
Website link	Check if a URL link is found in the surrounding tokens
Email	Check if a sample email is found in the surrounding tokens
Date	Check if a date in any format is found in the surrounding tokens
Similarity	Features of the similarity
URL	Check whether the token is similar to an URL in the page
Part of Email	Check whether the token is similar to an email on the page

Figure 2.2.2.2.2.1. The list of manually selected features

2.2.2.2.3. Dictionaries

Dictionaries are compiled and stored list of word and special characters. Dictionaries can be included in natural language processing tasks and figure as a feature by representing either the presence or the absence of a token in these dictionaries. In entity detection related tasks dictionaries help to leverage and bind word to defined entity category. In fact, Gazetteer lookup was mentioned in name entity recognition challenges as a way to increase precision and reduce ambiguity [31]. Dictionaries can be approached in two different fashions. The first approached is the standard language dictionary. This dictionary is the compilation of words in a certain language along with the definitions. Linux dictionary and WordNet are examples of this approach. A second alternative is to construct a gazetteer list. A gazetteer list is a collection of words or group of words stored as a dictionary. However, gazetteer lists are different from standard dictionaries since

they contain only nouns especially a preselected list of well-known names of entities. It includes places, countries, companies, people etc... The two alternatives were included as lookup features in the analysis. This feature can quickly detect the presence of organization names and easily disambiguate some tokens to get the status of a company name.

2.2.3. Feature Encoding

The process of the feature engineering provided means to detect important patterns for the analysis. The extraction of features allows to capture possible indicator of the company title name. The feature extraction resulted with a total of 82 features for each individual token. The features as seen in the above section are presented into Boolean format as 0 or 1, as integer value such as the count or the length attributes and string value. String and category attributes demand and encoding process before they are sent to machine learning classifiers. The encoding method applied in this analysis is the the one hot encoding. In this case a unique dimension is assigned for each feature. For example, when considering a bag-of-words representation over a vocabulary of 40,000 items, x will be a 40,000-dimensional vector, where dimension number 23,227 (say) corresponds to the word dog, and dimension number 12,425 corresponds to the word cat. A document of 20 words will be represented by a very sparse 40,000-dimensional vector in which at most 20 dimensions have non-zero values. Correspondingly, the matrix W will have 40,000 rows, each corresponding to a particular vocabulary word. When the core features are the words in a 5 words window surrounding and including a target word (2 words to each side) with positional information, and a vocabulary of 40,000 words (that is, features of the form word-2=dog or word0=sofa), x will be a 200,000-dimensional vector with 5 non-zero entries, with dimension number 19,234 corresponding to (say) word-2=dog and dimension number 143,167 corresponding to word0=sofa. This is called a one-hot encoding, as each dimension corresponds to a unique feature, and the resulting feature vector can be thought of as a combination of high-dimensional indicator vectors in which a single dimension has a value of 1 and all others have a value of 0.

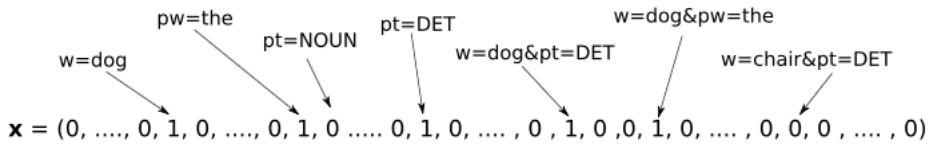


Figure 2.2.3.1 Example of feature encoding into unique dimensions

2.3. Prediction Methods

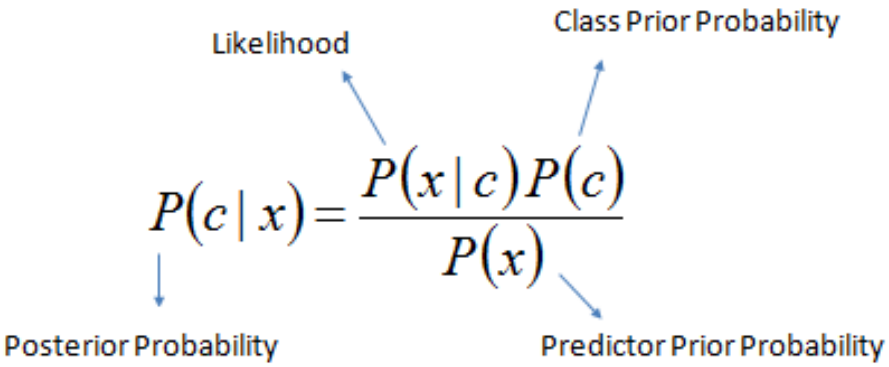
The company name detection results into a classification task. The prediction of the company name is binary classification process where the analysis will consist of predicting whether a word is a company name for the web page or not. The prediction is achieved at two levels. The first level is at the token level where each token is shown a prediction demonstrating the probability of resulting as a company name and another level which is at the page at the page level, the prediction consists of detecting the token with the highest likelihood to figure as the company name of the webpage. At the token based level features are extracted and the classification algorithm has been applied.

2.3.1. Classification Methods

The outcome of the feature engineering and the cleansing of the data allows to reduce the task into a machine learning problem. The detection can be transform into a classification problem with regular machine learning algorithm and model training. The following algorithm were used in the binary classification of the words at the token level approach of the analysis.

2.2.1.1. Naive Bayes

Naive Bayes is a machine learning algorithm extracted from the probabilistic theorem of Bayes. It is recognized as the Naive Bayesian model in the field of artificial intelligence. The model is simple and easy to build but relies on an assumption of independence between features predictors. It uses the formula illustrated below.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 2.2.1.1.1. Naïve Bayes formula [42]

Naive Bayesian model is considered as a generative model and the probability of each class is computed to compare the likelihood of belonging to a specific class. It takes the prior knowledge and computes the probability of the hypothesis given.

2.2.1.2. Decision tree

Decision tree is a machine learning procedure based on the entropy of the samples. A decision tree is a supervised algorithm and is well used for the simplicity of the algorithm. They can serve as classification method or a regression predictor regarding the problem. It starts from a root node and based on the entropy the data the samples are split until it reaches a leaf node. In our work, we have implemented this algorithm by means of a third party framework.

2.2.1.3. Random Forest

Random Forest classifier is an example of assembling learning. Assemble learning is the implementation of various learning strategies into a single process in order to bring a solution to a problem in contrast to early machine learning methods which works on a single learning method to solve the problem. Assembler methods usually give a better generalization that single learner method [43]. It is robust again overfitting, outliers and helps to boost weak learners. Random forest is an assemble learning with decision tree as base learners. The figure below illustrates and example of random forest.

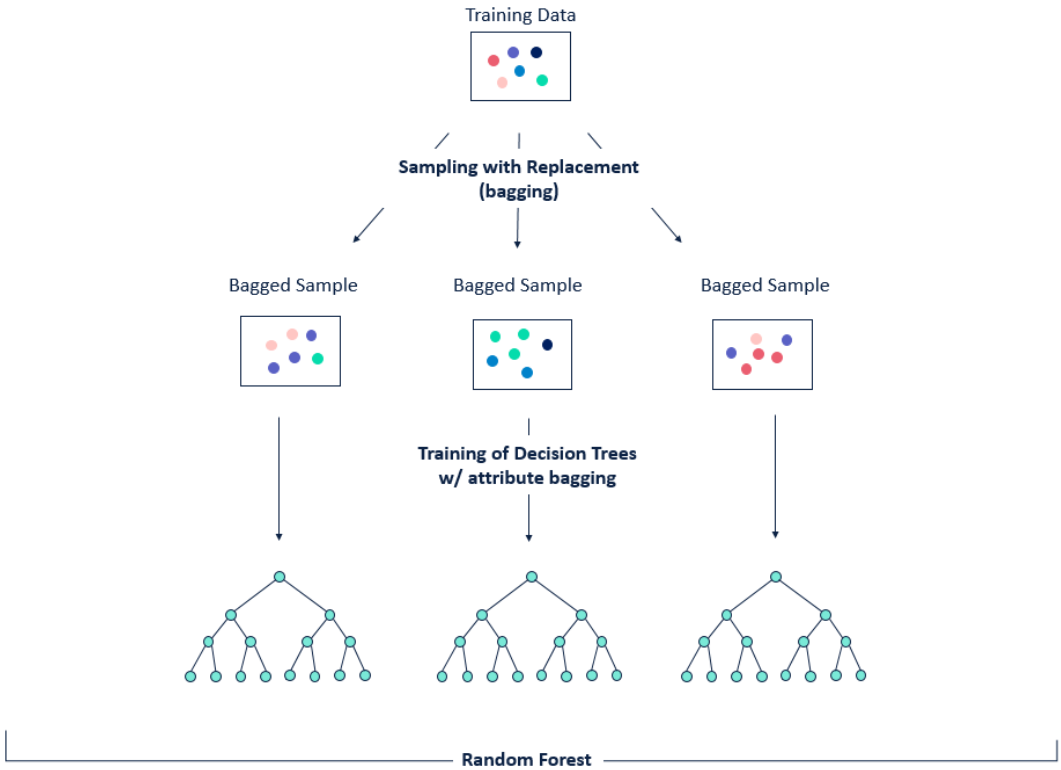


Figure 2.2.1.3.1. An example of random forest classification [56]

2.2.1.4. Conditional Random Field

Conditional Random Field (CRF) is a probabilistic discriminant model introduce by Lafferty et al. [38] for the prediction of sequential and time series data. It was inspired by the hidden Markov models and aimed at providing better accuracy in sequential predictions It has been widely used in the field of natural language processing where it revealed excellent results. CRF allows classifying an observation x which usually is a sequence of units regarding their labeled y. The equation is presented in the figure below.

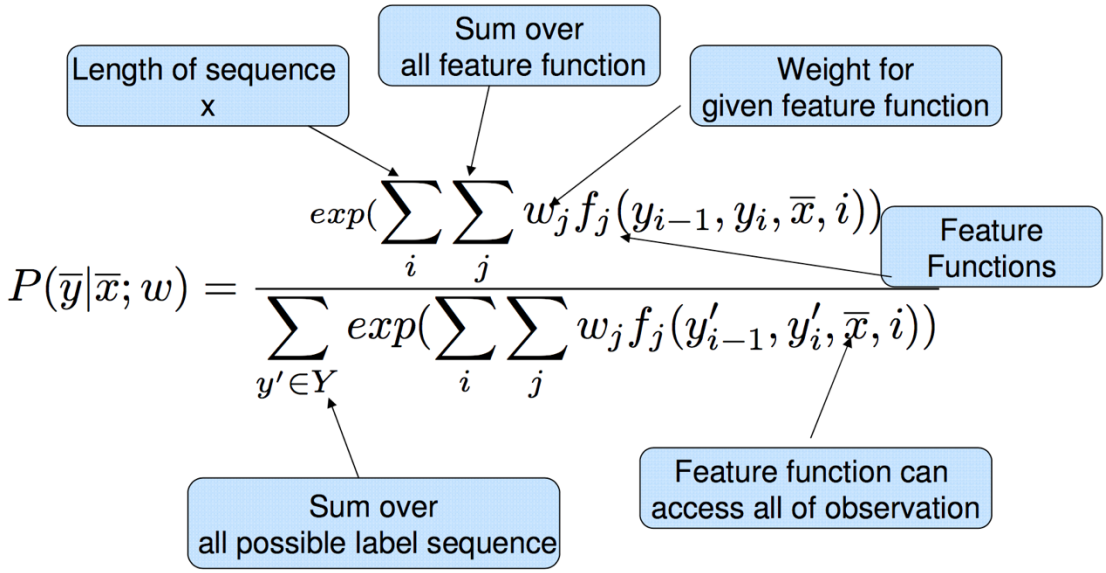


Figure 2.2.1.4.1. The statistical formula of CRF [44]

2.2.1.5. Neural Networks

Neural Networks or well known as Artificial Neural Networks appeared recently as a powerful learning method such that the applications are observed in image recognition, speech analysis, and text processing as well. However, the early appearance goes back to the middle of the 20th century with psychologists who tried to understand the human brain. In fact, the neural network uses human to process and transfer information. Neural networks consist of layers and each layer is comprised of interconnected nodes which in return contain an activation function. There are three main layers in a neural network. Figure 2.2.1.5.1 represents an example of neural network. The first layer called the inputs layer where the data is feed to the network. A second layer which is called the hidden layer(s) which is composed of one or more layer and is the layer where the actual processing is demonstrated via a process of weighed the links between nodes. There are connections from the inputs nodes to the hidden layer and from the hidden layer to the last layer. The last layer is the output layer which provides the final response of the network. The signals forwards through the networks and the errors are sent back to correct and adjust the weight coefficient through a dynamic algorithm known as backpropagation.

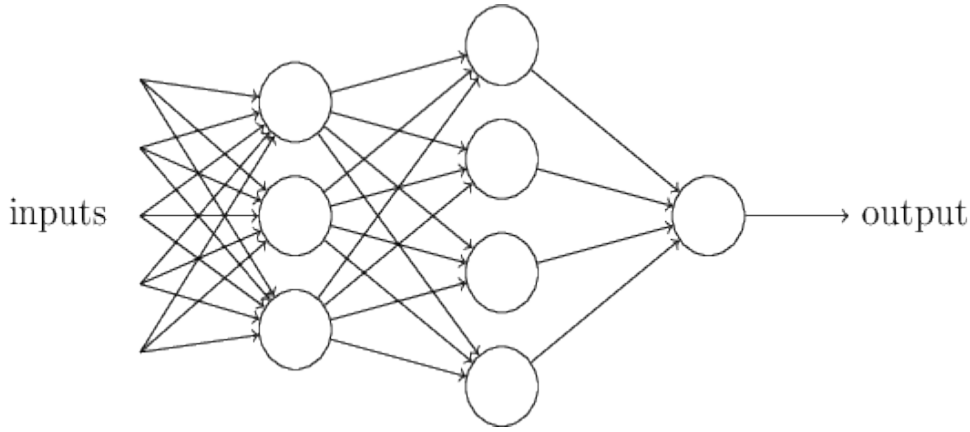


Figure 2.2.1.5.1. A based architecture of a Neural Network [41]

2.2.1.6. Support Vector Machine

Support vector machine is a classifier with the goal of extracting a pattern from complex data. The main idea behind the support vector machine is to determine a separator which maximizes the distance between support vectors. It tries to find the best hyperplane that maximizes the margin distance. The figure below reflects a summary of a support vector binary classification.

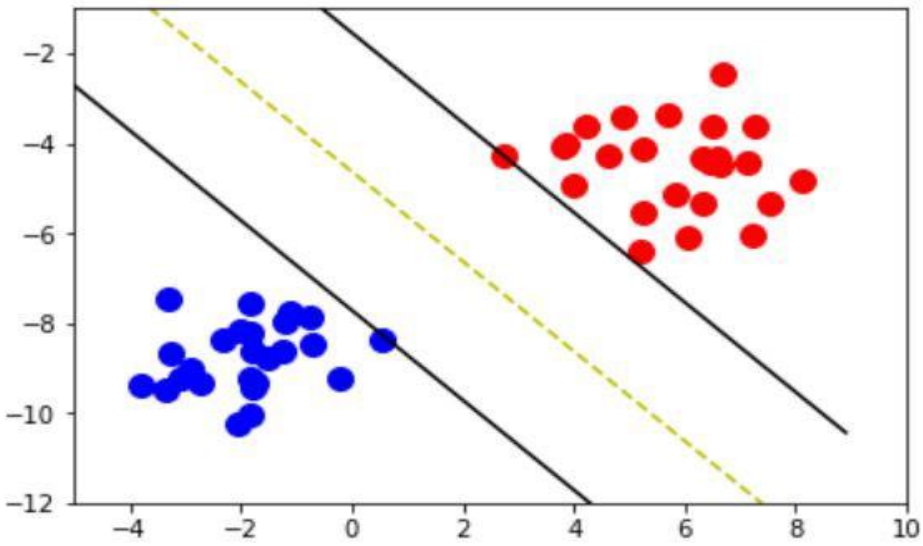


Figure 2.2.1.6.1 Support Vector Machine [45]

2.3.2. Rule-Based Methods

The experiment exploited adequate rules to predict the company name in a separate model. The dataset beyond the content of the web page also contained the title of the web page and the Universal Resource Locator (URL) as a separate column. In order to leverage the prediction and utilize these pieces of information retrieved by the crawlers, a rule-based method was attempted. The rules were based on the similarities between the domain URL and the token of the title and the body content.

2.3.2.1. Regular Expression

Regular Expressions (RE) are pattern matching methods. There are a series of characters to represent and lookup for a specific match. The pattern is used to search for words or group of words which fit the rules. Regular Expressions are flexible and the characters can be made of numbers, string, and special characters.

RE	Expansion	Match	First Matches
\d	[0-9]	any digit	Party␣of␣5
\D	[^0-9]	any non-digit	Blue␣moon
\w	[a-zA-Z0-9_]	any alphanumeric/underscore	Daiyu
\W	[^\w]	a non-alphanumeric	!!!!
\s	[␣\r\t\n\f]	whitespace (space, tab)	
\S	[^\s]	Non-whitespace	in␣Concord

Figure 2.3.2.1.1. Example of Regular Expression [29]

2.3.2.2. Common Similarity

This function highlights the number of characters in the intersection between two strings. Given a string A with n characters and a string B with n characters, it finds the common characters appearing in both string A and string B regardless of the order of characters. It identifies and counts the characters observed in string A and present in string B.

2.3.2.3. Minimum Edit Distance

The differences between the two words can be measured using the edit distance. The edit distance is a metric for measuring the similarity two strings. It does not only rely on the common characters shared between the strings but also checks the spelling differences of their characters. It is defined as the number of editions required to convert

a string A to string B. It counts operations such as substitutions, insertions, deletions necessary for the transformation. An example of calculation is represented below.

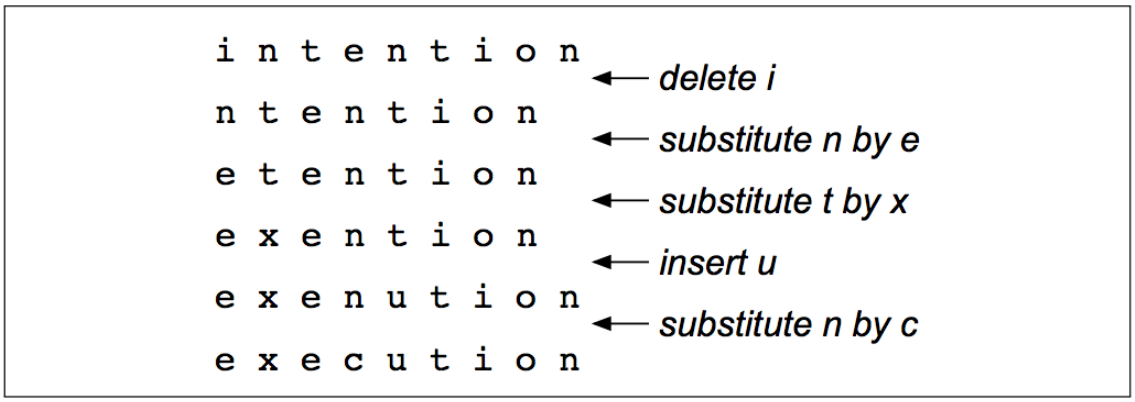


Figure 2.3.2.3.1. Demonstration of edit Distance computation [29 chapter 2]

2.4. Tools

The experiments are implemented from scratch in a python environment. Python is a powerful language with a syntax suitable for the development of algorithms. The language is well supplied of built in functions and furthermore allows an easy integration of third party libraries through the package management system. These properties make the language a preferable option for machine learning and text processing experiments. In fact, python is listed as the most used language in artificial intelligence projects [46]. The programming built in functions are utilized for standard operation, however special packages were included to make use of existing package and save time for the implementation.

2.4.1. Natural Language Toolkit

Natural Language Toolkit (NLTK)is a popular platform for text processing written in python [47]. The library is open sourced and incorporates linguistics means to text analysis. NLTK contains a large variety of methods such as sentence segmentation, word tokenization of the, parses tree generation, group chunking, part of speech tagging, name entity Recognition and various corpora for training and testing. The package was employed for tokenization of the web page content and the extraction of simple features and cascading features such as the part of speech tagging.

2.4.2. Spacy

Spacy is another text mining package designed for production ready systems [48]. Unlike NLTK or other libraries for natural language processing, spacy offers an already trained model for real life data without a training process with your own data. The framework was introduced in order to meet the requirement of the production system. The model is trained my means of deep learning algorithm with clean datasets and optimized to provide the best results to unseen data. NLTK has some limits. In our analysis, NLTK was not able to extract features such as the semantic roles labelling and the reference relation. As a results, spacy was used to identify the remaining features.

2.4.3. Scikit-learn

Scikit-learn is an efficient framework for text and data analysis [49]. It was developed to incorporate machine learning framework into a simplified and productive environment. Scikit-learn contains techniques and algorithm required for the preparation and the training of a model. Algorithm put into practice in the prediction of the company name were trained using Scikit-learn classifiers. Subsequently to the features extraction, the data is handed over to the Scikit-learn package for the classification.

Chapter 3

Experiments and Analysis

The assessment of this experiments is evaluated at the token level and the page level. Unlike regular training methods the evaluation is performed at two level. A token level referred as a locally based prediction, and a page level referred as a level prediction.

3.1. Local Based Prediction

The local based prediction is the prediction per-token basis. The prediction is a binary classification of the words of the page. Each token from the page receives an indicator of the class it belongs. Scores are computed by the classifier and the token is determined as a company name or not. The accuracy of this level is measured with standard metrics including the overall accuracy, precision, recall, F-scores, and AUC scores.

3.1.1. Metrics

3.1.1.1. Accuracy

Accuracy is the general score assigned to the classification method. The score is computed using the classification output against the desired output of the data employed in the analysis. It counts the number of labels classified right compared to the dataset labels. The formula is presented as below.

$$\mathbf{Accuracy} = \mathbf{100} \times \frac{(\mathbf{TP} + \mathbf{TN})}{(\mathbf{TP} + \mathbf{FN} + \mathbf{FP} + \mathbf{TN})} \qquad (3.1.1.1.1)$$

3.1.1.2. Precision

Precision reflects the number of positives predictions are indeed correctly classified positive by the algorithm. Precision checks the score of the positive labels against the overall positivity score.

$$\textbf{Precision} = 100 \times \frac{(\textbf{TP})}{(\textbf{TP} + \textbf{FP})} \quad (3.1.1.2.1)$$

3.1.1.3. Recall

The recall is the computation of the actual positives the classifier captures though classifying it as a positive label. It is well recognized in the field of machine learning as a sensitivity score or the rate of the true positive. The summary of the computation is represented as follows:

$$\textbf{Sensitivity} = 100 \times \frac{(\textbf{TP})}{(\textbf{TP} + \textbf{FN})} \quad (3.1.1.3.1)$$

3.1.1.4. F Score

F-Score is a measure which seeks to balance the recall and precision metrics-score computes a computes an average known as the harmonic mean of the precision and the recall. The equation is formulated as follows.

$$\textbf{F} - \textbf{measure} = 2 \times \frac{(\textbf{Precision} \times \textbf{Recall})}{(\textbf{Precision} + \textbf{Recall})} \quad (3.1.1.4.1)$$

3.1.1.5. AUC

AUC basically stands for Area under the ROC Curve. The Roc (Receiver Operating Characteristics) curve is a visualized graph depicting the classification model at different thresholds. AUC shows the area below the ROC curve and illustrates the measure of separability.

3.1.2. Class Imbalance Problem

The accuracy on the dataset was high but a close analysis of the precision and the recall metrics revealed a low percentage especially for the company name class. This brought out a new issue in the experiment which is faced in most machine learning

problems [50]. The issue is recognized as the class imbalance problem. It occurs when the classes are not assembled equally. A specific class overweighs and is represented more frequently than other classes. In our analysis, the classification problem is imbalanced such that the label of company names are a minority compared to the none company name class. To reduce the impact of the class imbalance on the model two main alternatives are proposed in the literature. The first is under sampling and the other is oversample.

3.1.2.1. Under Sampling

Under sampling is a simple technique of reducing the dataset to sample having an equal number of sample per class. The samples of the majority class are reselected to the size of the minority class. The selection is implemented in multiples techniques. The criteria for selection in this analysis is based on the distribution of the samples and the selection is done in a bootstrap manner [51].

3.1.2.2. Over Sampling

Oversampling consists of enlarging the dataset to set a balance between the classes. The dataset is augmented such the minority class reached a with the majority class. New instances of the minority class are generated to compensate the inferiority. There are many algorithms which offered the possibility of generating synthetic samples from an imbalanced dataset. The most popular is the Synthetic Minority Over-Sampling Technique SMOTE [52]. In our experiment, this technique is made use of in order to increase the size of the samples of the company names. Oversampling provided better results than the under sampling technique.

3.1.3. Results

The results of the classifiers are presented in the following sections. There are two classes in the prediction of the tokens. Class A is the category of the company name class. They are the tokens suggestible of depicting the company name. Class B is the category of other than the company name. Any other token predicted is classified in this category.

3.1.3.1. Naive Bayes

We have trained the token-based classification with a multinomial naive Bayes version of scikit-learn. This algorithm is well used in text related experiments [53]. The results of the implementation are illustrated in the following tables.

Methods	Accuracy	Precision	Recall	F1 Score	AUC
Normal	98.55	70.80	59.56	63.02	59.57
Undersample	91.88	92.4	91.87	92.13	91.88
Oversample	96.81	96.99	96.81	96.81	96.81

Table 3.1.3.1.1 The average measures of the multinomial naive Bayes classifier produced at the token level classification

Methods	Precision		Recall		F1_measure	
	Class A	Class B	Class A	Class B	Class A	Class B
Normal	42.71	98.90	19.50	99.63	26.78	99.26
Undersample	87.70	97.10	97.42	86.33	92.56	91.71
Oversample	94.08	99.90	99.91	93.71	96.91	96.71

Table 3.1.3.1.2. The results of the multinomial naive Bayes for each class in the Binary classification at the token level.

3.1.3.2. Decision Trees

Decision trees are implemented with the scikit-learn packages where the parameters are the standard pre optimized offered by the framework [54]. The default parameters are constructed with a max depth of the tree to none which means node are expanded until all leaves contain less than the minimum sample split and the mean sample split is set to 2. The minimum sample at the leaf is 1. The results of the implementation are illustrated in the following tables.

Methods	Accuracy	Precision	Recall	F1 Score	AUC
Normal	99.83	98.12	95.76	96.90	95.76
Undersample	95.03	95.03	95.02	95.02	95.03
Oversample	99.78	99.78	99.78	99.78	99.78

Table 3.1.3.2.1. The average measures of the Decision Tree classifier produced at the token level classification

Methods	Precision		Recall		F1_measure	
	Class A	Class B	Class A	Class B	Class A	Class B
Normal	96.36	99.88	91.57	99.95	93.90	99.91
Undersample	95.11	94.95	94.94	95.11	95.02	95.03
Oversample	99.74	99.82	99.82	99.74	99.78	99.78

Table 3.1.3.2.2. The results of the Decision Tree for each class in the Binary classification at the token level

3.1.3.3. Random Forest

The application of random forest model went through an optimization step. The parameters were selected after a grid search of the optimum parameters. The grid is composed of the number of trees of the algorithm starting from 5, 10, 20, 50, 100, 200, 300, 400, 500, and 1000. The search was done with a 10 fold cross-validation. The best parameters obtained respectively regarding the depth and the number of estimators are 400 and 300. The results are presented as below.

Methods	Accuracy	Precision	Recall	F1 Score	AUC
Normal	99.67	99.77	88.2	93.20	88.20
Undersample	94.23	94.35	94.2	93.20	94.23
Oversample	99.88	99.88	99.87	99.88	99.88

Table 3.1.3.3.1. The average measures of the Random Forest classifier produced at the token level classification

Methods	Precision		Recall		F1_measure	
	Class A	Class B	Class A	Class B	Class A	Class B
Normal	99.88	99.67	76.41	99.99	86.58	99.83
Undersample	92.07	96.63	96.80	91.65	95.95	94.14
Oversample	99.96	99.80	99.79	99.96	99.88	99.88

Table 3.1.3.3.2. The results of the Random Forest for each class in the Binary classification at the token level.

3.1.3.4. Multi-Layer Perceptron

The model was trained with a Multi-Layer Perceptron architecture. Due to the size of the tokens in the dataset and the available hardware resources, the multi-layer perceptron model did not receive a deep optimization of the parameters. The default used parameters are 2 hidden layers of size 5 and BFGS as the error optimization technique. The batch size is set to auto with a minimum of 200.The learning rate is 0.001 and the regularization parameter L2 is set to 0.0001.

Methods	Accuracy	Precision	Recall	F1 Score	AUC
Normal	99.64	94.23	92.19	93.18	92.19
Undersample	96.73	96.72	96.72	96.72	96.73
Oversample	98.76	98.76	98.75	98.75	98.76

Table 3.1.3.4.1. The average measures of the Multi-Layer Perceptron classifier produced at the token level classification

Methods	Precision		Recall		F1_measure	
	Class A	Class B	Class A	Class B	Class A	Class B
Normal	88.69	99.78	84.54	99.85	86.56	99.81
Undersample	96.56	96.89	96.90	96.55	96.73	96.72
Oversample	98.53	98.99	98.99	98.52	98.76	98.75

Table 3.1.3.4.2. The results of the Multi-Layer Perceptron for each class in the Binary classification at the token level.

3.1.3.5. Support Vector Machine

The training of the support vector machine was expensive in time. As a result, the model of the support vector machine did receive any optimization of the parameters. The standard parameters were applied in the training [55]. The default parameters are composed of a penalty C of 1 and a kernel of RBF. The standard degree is assigned with 3 and a coefficient of 0.0. The tolerance for stopping is 0.001 with a cache size of 200. Moreover, the complexity of the algorithm and the size of the data did not allow us to train entirely the SVM model. As a results a seventh of the data was used for in the SVM model. The results of the implementation are illustrated in the following table.

Methods	Accuracy	Precision	Recall	F1 Score	AUC
Normal	98.53	49.26	50.00	49.63	50.00
Undersample	56.39	59.56	56.48	52.62	56.48
Oversample	84.19	84.40	80.26	84.16	84.19

Table 3.1.3.5.1. The average measures of the Support Vector Machine classifier produced at the token level classification

Methods	Precision		Recall		F1_measure	
	Class A	Class B	Class A	Class B	Class A	Class B
Normal	0.00	98.53	0.00	100.00	0.00	99.26
Undersample	65.15	53.97	28.10	84.86	39.26	65.98
Oversample	87.11	81.70	80.26	88.12	83.54	84.79

Table 3.1.3.5.2 The results of the Support Vector Machine for each class in the Binary classification at the token level.

3.1.3.6. Conditional Random Field

Conditional Random Field provides perfect results for sequential classification tasks. However, this algorithm was not able to capture the patterns in this analysis. The results are not as good as standard classifiers. It brings about a good point in the analysis. Although the words are sequential, the company name detection task is a binary classification task. Regular binary classifiers outperform sequential classifiers in the process. The results for the Conditional Random Field are shown below.

Methods	Accuracy	Precision	Recall	F1 Score
Normal	97.56	97.50	98.80	98.10

Table 3.1.3.6.1. The average measures of the Conditional Random Field classifier produced at the token level classification

Methods	Precision		Recall		F1_measure	
	Class A	Class B	Class A	Class B	Class A	Class B
Normal	0.00	98.80	0.00	99.99	0.00	99.40

Table 3.1.3.6.2. The results of the Conditional Random Field for each class in the Binary classification at the token level.

3.2. Global Based Prediction

At this predicting stage, a single token is selected to a candidate for the company name for the page. Global based prediction combines all the prediction from the token based to filter the best title for the page. It discards tokens predicted as not candidate and a coefficient to each candidate for ranking criteria. The classification probability issued from the token based along with the frequency of the token predicted is associated to make up the ranking coefficient.

3.2.1. Post processing

Subsequent to the local prediction, the token marked as the possible company name is inserted into a post-processing filter. The filters take care of removing false predictions by the token based classification. Some characteristics are determinative of a word not reflecting the company the webpage. These characteristics include the length of the word very long (30) or very short (1), the word represented in the English dictionary as normal words (verb, conjunction) and the entity classified as other than a named entity.

3.2.2. Similarity Scores

The final prediction is evaluated from the similarity to the manually labeled company name. In general, some words although are not strictly similar to the company name can still be as a company and deserved to be labeled as a right prediction. This is applied in the field of title extraction [15]. In our experiment, we have adopted two evaluation measures. The first is represented as below.

$$\mathbf{Similarity(t1, t2)} = \frac{\mathbf{d(t1,t2)}}{\mathbf{max\ (l1,l1)}} \qquad \mathbf{(3.2.2.1)}$$

The output of this function indicates the affinity between two words. In our experiment, the first word is the prediction (t1) from the global classifier and the second input is the manual label of the company name(t2). The top part is the minimum edit distance for the two words and the bottom is the maximum length of words. The input l1 and l2 are respectively the lengths of t1 and t2. The max is the max length of the two

inputs. In order to make a selection from the acceptable prediction, a threshold has been applied. The threshold is a coefficient of 0.3. This coefficient has been selected regarding the general literature methods [21]. The goal of this measure is to provide an acceptable margin to discard prediction candidates.

The second classification measurement is the dice coefficient [22]. The dice coefficient measure also the similarity between the two words. The difference between the formula 3.2.2.1 and the Dice coefficient is that the first gives an edition score whereas the second give a percentage score. Dice coefficient output is a percentage score and is averaged to give a general score the overall corpus. It computes the weight of adjacent pairs characters of both words.

$$\mathbf{Similarity(t1, t2)} = \frac{\mathbf{2 \times |\, pairs(t1) \cap pairs(t2)|}}{\mathbf{|\, pairs(t1) \cup pairs(t2)|}} \qquad \mathbf{(3.2.2.2)}$$

A pair is a 2-gram, a combination of two characters within a word. A pair (t1) is a sequence of words composed of subsets of two characters. The subsets are selected consecutively regarding the order of characters in the tokens. It calculates the pairs in intersection against the pairs shared between tokens. This algorithm is chosen for the language attributes it holds. This algorithm is language independent [22] and robust concerning the order affability of the tokens.

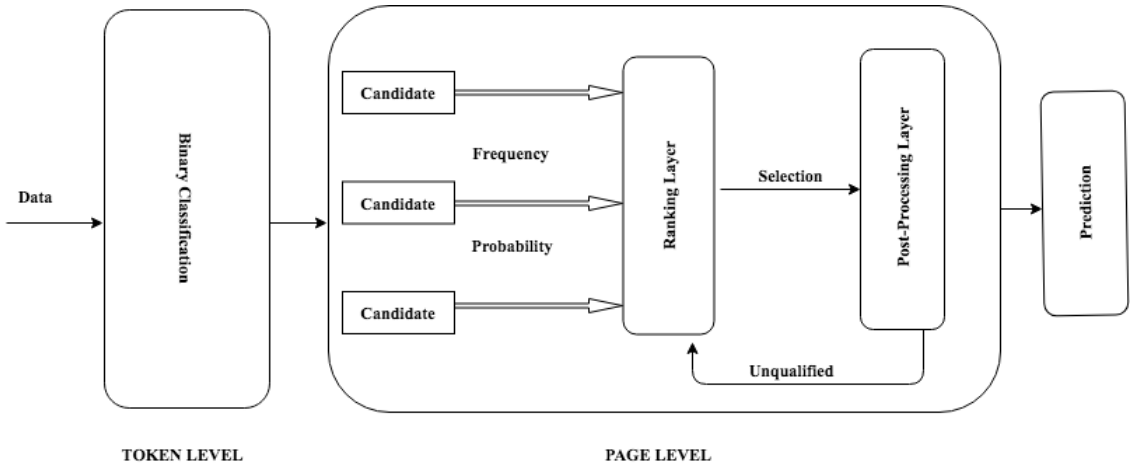


Figure 3.2.2.1 Full architecture of the process

3.2.3. Results

There are three classes considered in the measurement. The first class defines the correct prediction of the company name. The second is the undefined class which occurs in cases where the prediction derives an empty token as the company name. This was not included in the training but such cases are likely to happen in a real production environment. The last class represents the percentage of the wrong prediction issued by the classifier. The Dice score is average percentage of the Dice coefficient for the right predictions. The global prediction outcomes are illustrated in following tables.

Methods	Accuracy			Dice score
	Right	Undefined	Wrong	
Normal	20.26	71.85	07.89	96.77
Undersample	58.16	0.0	41.83	94.20
Oversample	60.13	0.0	39.86	96.83

Table 3.2.3.1. The results generated by the Naïve Bayes classification at the global level prediction of the page.

Methods	Accuracy			Dice score
	Right	Undefined	Wrong	
Normal	72.54	22.24	05.22	99.47
Undersample	56.86	0.0	43.13	94.77
Oversample	73.20	10.47	16.33	97.43

Table 3.2.3.2 The results generated by the Decision Tree classification at the global level prediction of the page.

Methods	Accuracy			Dice score
	Right	Undefined	Wrong	
Normal	64.05	34.65	01.30	98.66
Undersample	58.16	0.0	41.83	95.60
Oversample	74.50	21.58	03.92	99.56

Table 3.2.3.3 The results generated by the Random Forest classification at the global level prediction of the page.

Methods	Accuracy			Dice score
	Right	Undefined	Wrong	
Normal	73.20	15.04	11.76	92.85
Undersample	62.74	0.66	36.60	94.39
Oversample	69.28	3.27	27.45	96.22

Table 3.2.3.4 The results generated by the Multi-Layer perceptron classification at the global level prediction of the page.

Methods	Accuracy			Dice score
	Right	Undefined	Wrong	
Normal	23.33	0.0	76.67	100.0
Undersample	31.81	0.0	68.19	100.0
Oversample	53.33	0.0	46.67	91.01

Table 3.2.3.5 The results generated by the Support Vector Machine classification at the global level prediction of the page.

3.3. Rules Base Approaches

To make use of all the data retrieved by the web crawler , a special field was included in the analysis which is the domain URL. The domain URL could serve to predicate the company title name and approach the analysis differently. A set of rules could be generated in the context of the base in order to extract to company name of the page.

3.3.1. Search methods

The dataset given contain the domain name of the company. A column was populated with URL of the pages crawled. After a deep analysis of the relationship the company name, we judge convenient to make use of the domain name in the analysis. In general, the company name is in most cases similar to the domain name, therefore, we proposed a machine learning model and a search and ranking methods to support and boost the results obtained.

3.3.2. Rank Approach

Search is a magnificent way to retrieve some data. Given the domain URL, a simple search was implemented. This search takes care of parse through all tokens of the page and compares each token to the domain name of the page. The comparison is achieved a score is assigned to the token. The score calculated with the minimum edit distance and the common characters shared between the token and the domain name. The final score is the difference of the minimum edit distance and the count common characters. The pages were parse into a tokenizer and the similarity were prediction for each token of the web pages. The tokens are sorted by their coefficient score and the highest ranked is classified as the company name title. The results of this procedure is listed in the table below (3.3.3.1).

3.3.3. Model Approach

The search and rank approach produces satisfying results. However, a machine learning alternative is proposed to infer context-based features. Features have been

extracted from the text content and the title of the web page. The model makes use of more attributes of the page with possibilities of pointing towards the company name of the web page. A pre-processing is introduced in order to select tokens with probabilities of representing the company name of the page. Tokens are compared to the domain name of the page. Tokens showing high dissimilarity from the domain URL were discarded reduce the candidate size. The features extraction task is only applied to tokens retained after the pre-processing phase. A model is trained using regular classification methods and the performance is measured with the above metrics (3.2.2.1) (3.2.2.2). The final list of features is illustrated as below.

Features	Attribute
Present in Title	Check if the token is found in the title
Present in content	Check if the token is found in the title
Abbreviation to the domain	Check if the token is an abbreviation or a short form of the domain
Count in Title	The number of appearance in the title of the page
Count in Content	The number of appearance in the content of the page
Abbreviation count	The number of abbreviation tokens
EditScore	The minimum edit score to the domain name
CommonScore	The count of the characters in common averaged by the token size
Length	The count of the characters
Tokens	The count of words in the token

Table 3.3.3.1. The list of feature extracted for the training of the model including the domain Name.

3.3.4. Pipeline Approach

This approach takes into account only the text of the page and the latter which includes the domain in the detection task. Issue from the results produced by these two proposed methods, a third alternative is derived. The third alternative seeks to combine the benefits of the two first suggestions. A pipeline is designed connecting the two models. The pipeline takes at the rear the prediction of the models trained with only the content and attempts the prediction of the second at the front in case prediction of undefined by the first model. Although, the first model has lower accuracy compare to second, it tends to be more precise in the detection of the company names. The final results are recorded in the table below in comparison with the attempted techniques.

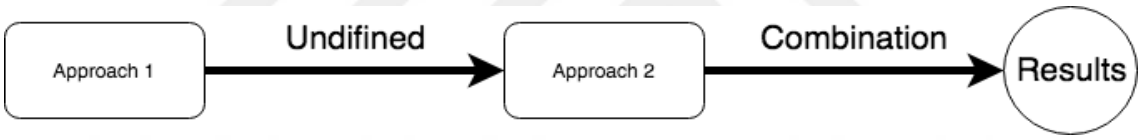


Figure 3.3.4.1 Overview of the pipeline architecture

Methods	Accuracy			Dice score
	Right	Undefined	Wrong	
Rules based	99.71	0.0	00.28	94.91
Model-based	96.75	00.28	2.97	95.96
Pipeline 1 Model-based + Rules based	98.01	0.0	1.98	96.12
Pipeline 2 Rules based + Model-based	99.71	0.0	00.28	95.90

Table 3.3.4.1 The Accuracy at the global based prediction for the rule based approaches and the pipeline prediction.

3.4. Discussions

From the results of the experiment, we operated a deep analysis on possible causes susceptible of inhibiting the performance. The errors were examined to find the origin of the miss prediction for both the machine learning model and the ruled based techniques. As a results the following aspect were identified to impact negatively the methods.

The language was at the core for most of the errors that lead to a low performance of the model. English is the main language of the focus and the study was built around the syntax and the language style of the English language. The dataset selected for the experiment was extracted meticulously to capture text characterized by words and sentences only enclose in English. For this purpose, tools and packages utilized were meant to extracted pattern and features according to the sentence structure of the English language and the word properties of English. Therefore, the model first pre-processes text from a new language and fails to extracted accurate features from sentences and each unit token and constrains the model to commit a wrong prediction for the content provided. As a result, the overall performance is reduced due to new languages are seen in the test set other than English.

The Domain name is also another subject that reduced the performance of the model. Unfortunately, the web is structured in such a way that domain URL are centred around English. Domain names are represented in the English language which disable the possibility to encode words from any other language into a domain format. A token written in Turkish as an example with a character not existing in the English alphabet scope will have to be somehow be converted the closest English's character or alternatively be removed to be represented in the domain URL. This study focuses on retrieving the company name with means of the domain get affected by this case. The domain name may consequently differ or have a huge contrast from the name represented in the context due to a possibility that the names are well recorded in the page in a language other than English and but transformed to a new word to make it compatible to a domain name. Hence makes difficult to detect patterns and similarities between the tokens of the pages and the domain name. The constraint of the domain URL to fit a

certain format may induce unrelated patterns between the company name and the domain and by the same way reduced the performance of the model prediction.

In addition, the presence of empty content impacted the model to make an accurate prediction. The dataset was generated by the process of crawling web page on around the web. Within this data appears some pages inconvenient to ensure a good prediction of the company name. There are essentially two types of this kind of pages. The first type is the empty content. The crawler just retrieved either an empty page from the website or was restricted of accessing the content of the webpage which in return delivered an empty content, basically a blank text. Another category of faulty content is text from pages under construction. Some web pages around the web are sometimes shut down due to various issues or the domain was purchased but a website was not constructed to serve on the domain request. In this state, the crawler captured some text that is not relevant to determine the company name, such as programming code or script written to redirect users in case or messages to address the state of the domain for the current situation. Defected content observed in the data affects negatively the model prediction and induce a lower accuracy in the overall prediction. The model fails to find a prediction in such case by labeling the page as undefined.

Chapter 4
Conclusions and Future Prospects

4.1 Conclusions

In this thesis, a novel machine learning method is developed to detect company name from a web page content. Several features are extracted that reflect the context and textual characteristics of the tokens. The proposed classifier has two stages. The first stage classifies whether tokens originating from the web page belong to a company name or not and the second stage employs a rule-based method that searches and ranks tokens based on similarity measures. The first stage obtains a lower accuracy but higher precision in comparison to the rule based learning method and the combination of the two methods into a single hybrid solution has demonstrated improved accuracy and precision.

4.2 Future Prospects

Several directions can be considered as future work. The dataset in the analysis was generated through a crawling process of random webpages and the content gathered resulted in noisy data, which necessitates a cleansing step either at the crawler html parser level or through a mechanism of discarding unwanted text from the web pages. In addition to the website text obtained from the crawler, the HTML source code can also be associated with the textual output of the crawler to allow feature extraction from the HTML tags. Moreover, additional web sites can be labelled in order to increase the number of training set samples. Data augmentation techniques can also be used for this purpose, which may bring further improvements in the accuracy. At the processing layer, a better architecture can be considered to allow a transfer learning between the token-based model training and the model that makes predictions at the website level. To reach this goal, the cost function can be designed in such a way that allows the final output error computed at the website level is used to update the parameters of the token-based model during training. Furthermore, hyper-parameters of the classifiers such as support vector machine and deep neural networks can be optimized better, which might have the potential to improve the prediction accuracy. Finally, an excellent alternative can be to cluster the websites and train separate models for each business area. In the present study,

the pages were assembled regardless of the business domain and the service field of the websites. Websites from the same business area share some similarity in the display of the pages and the text content. This similarity can improve the quality of features and boost the accuracy of the prediction models.

BIBLIOGRAPHY

[1] Radha Guha, " Exploring the Field of Text Mining," International Journal of Computer Applications (0975 – 8887), Volume 177 No.4, 11-17 (2017).

[2] https://www.ibm.com/blogs/business-analytics/data-is-everywhere/ (10.11.2018)

[3] Andreas Hotho, Andreas Nurnberger, Gerhard Paaß. , "A Brief Survey of Text Mining", LDV Forum - GLDV Journal for Computational Linguistics and Language Technology (20), *19-62* (2005)

[4] Vishal Gupta, Gurpreet S. Lehal, " A Survey of Text Mining Techniques and Applications," JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 1, NO. 1, 60-76 (2009).

[5] Amit Singhal, " Modern Information Retrieval: A Brief Overview," Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 24, No. 4, 35-42(2001).

[6] W. Bruce Croft Donald Metzler Trevor Strohman ,"Search Engines Information Retrieval in Practice" , Pearson Education, Inc, 2015.

[7] Dr.S.Vijayarani et al ," Preprocessing Techniques for Text Mining - An Overview," International Journal of Computer Science & Communication Networks,Vol 5(1),7-16, (2012).

[8] Diksha Khurana, Aditya Koli, Kiran Khatter, Sukhdev Singh, "Natural Language Processing: State of The Art, Current Trends and Challenges," DBLP:journals/corr/abs-1708-05148, abs/1708.05148, (2017).

[9] Eric Brill and Raymond J. Mooney, " An Overview of Empirical Natural Language Processing," AI Magazine Volume 18 Number 4 , 13-24 (1997).

[10] Georgios Petasis, Frantz Vichot , et al. " Using machine learning to maintain rule-based named-entity recognition and classification systems," ACL '01 Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, 426-433 (2001).

[11] Téllez-Valero, Alberto et al." A Machine Learning Approach to Information Extraction," Computational Linguistics and Intelligent Text Processing. CICLing 2005. Lecture Notes in Computer Science, vol 3406, 539-547(2005).

[12] More, Ajinkya. "Attribute Extraction from Product Titles in eCommerce." *CoRR*abs/1608.04670: n. pag. (2016)

[13] Azimjonov, Jahongir and Jumabek Alikhanov. “Rule Based Metadata Extraction Framework from Academic Articles.” (2018)

[14] Thadani, Kapil and Kathleen McKeown. “A Framework for Identifying Textual Redundancy.”, COLING '08 Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1, 873-880 (2008).

[15] Tanya Gupta. “Keyword Extraction: A Review.”, International Journal of Engineering Applied Sciences and Technology, 2017 Vol. 2, Issue 4, ISSN No. 2455-2143, 215-220(2017).

[16] Lin, Xiang et al. “Focused named entity recognition using machine learning.”, Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR 04, 281-288 (2004).

[17] Chen, Wang et al. “Title-Guided Encoding for Keyphrase Generation.” *CoRR*abs/1808.08575 (2018): n. pag.

[18] Kosala, Raymond and Hendrik Blockeel. “Web Mining Research: A Survey.” *SIGKDD Explorations* 2 (2000): 1-15.

[19] Costin Bădică,Amelia Bădică, " Rule Learning for Feature Values Extraction from HTML Product Information Sheets," Rules and Rule Markup Languages for the Semantic Web. RuleML 2004. Lecture Notes in Computer Science, vol 3323, 37-48 (2004).

[20] Dayne Freitag, " Information extraction from HTML: application of a general machine learning approach," AAAI '98/IAAI '98 Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, 517-523(1998).

[21] Xue, Yewei et al. “Web page title extraction and its application.” *Inf. Process. Manage.*, volume 43,1332-1347(2007)

[22] Gali, Najlah and Pasi Fränti. “Content-based Title Extraction from Web Page.”, 12th International Conference on Web Information Systems and Technologies, 204-210 (2016).

[23] Klampfl, Stefan and Roman Kern. “Machine Learning Techniques for Automatically Extracting Contextual Information from Scientific Publications.” , Semantic Web Evaluation Challenges. SemWebEval 2015. Communications in Computer and Information Science, vol 548, 105-116 (2015).

[24] Weerasooriya, Tharindu et al. “KeyXtract Twitter Model - An Essential Keywords Extraction Model for Twitter Designed using NLP Tools.” *CoRR* abs/1708.02912, n. pag(2017):.

[25] Luhn, Hans Peter. “The Automatic Creation of Literature Abstracts.” *IBM Journal of Research and Development* 2, Volume 2 Issue 2, 159-165(1958).

[26] Odena, Augustus et al. “Realistic Evaluation of Semi-Supervised Learning Algorithms.” (2018).structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions, J. Mol. Biol., 268, 209–225, (1997).

[27] Chapelle, Olivier et al., “Semi-Supervised Learning,” MIT Press (2006).

[28] Anjali M, Babu Anto, " Ambiguities in Natural Language Processing," International Journal of Innovative Research in Computer and Communication Engineering, Vol.2, Special Issue 5,392-394 (2014).

[29] Jurafsky, Daniel and James H. Martin. “Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Second Edition.”, Pearson Prentice Hall, (2008).

[30] https://arxiv.org/html/1708.09230v3 (02.11.2018).

[31] Ratinov, Lev-Arie and Dan Roth. “Design Challenges and Misconceptions in Named Entity Recognition.”, Proceedings of *the* thirteenth Conference on Computational Natural Language Learning, CoNLL '09,147-155 (2009).

[32] Mao, Xinnian et al. “Using Non-Local Features to Improve Named Entity Recognition Recall.” , Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation, 303–310 (2007).

[33] Kapociute-Dzikiene, Jurgita et al. “Exploring Features for Named Entity Recognition in Lithuanian Text Corpus.”, Proceedings of the 19th Nordic Conference of Computational Linguistics,vol.85,73-88(2013).

[34] Brill, Eric. “A Simple Rule-Based Part of Speech Tagger.”, ANLC '92 Proceedings of the third conference on Applied natural language processing, 152-155(1992).

[35] Collobert, Ronan et al. “Natural Language Processing (almost) from Scratch.” *Journal of Machine Learning Research*, Volume 12, 2493-2537(2011).

[36] https://medium.com/@gianpaul.r/tokenization-and-parts-of-speech-pos-tagging-in-pythons-nltk-library-2d30f70af13b (05.10.2018)

[37] http://www.nltk.org/api/nltk.tag.html (05.10.2018)

[38] Lafferty, John D. et al. “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.” *, Proceedings of the 18th International Conference on Machine Learning* , 282-289 (2001).

[39] Young, Tom et al. “Recent Trends in Deep Learning Based Natural Language Processing [Review Article].” *IEEE Computational Intelligence Magazine*, Volume 13, 55-75(2018).

[40] Villodre, Lluís Màrquez i et al. “Semantic Role Labeling: An Introduction to the Special Issue.”, *Computational Linguistics, Volume* 34, 145-159(2008).

[41] http://neuralnetworksanddeeplearning.com/chap1.html (05.10.2018)

[42] http://uc-r.github.io/naive_bayes (05.10.2018)

[43] Rokach, Lior. “Ensemble-based classifiers.”, *Artificial Intelligence Review, Volume* 33 ,1-39(2009).

[44] http://www.davidsbatista.net/blog/2017/11/13/Conditional_Random_Fields/ (05.10.2018)

[45] https://medium.com/deep-math-machine-learning-ai/chapter-3-support-vector-machine-with-math-47d6193c82be (05.10.2018)

[46] https://insights.stackoverflow.com/survey/2018/#technology (10.10.2018)

[47] https://www.nltk.org/ (10.10.2018)

[48] https://spacy.io/ (10.10.2018)

[49] https://scikit-learn.org/stable/index.html (10.10.2018)

[50] https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/ (05.10.2018)

[51] https://imbalanced-learn.readthedocs.io/en/stable/under_sampling.html (05.10.2018)

[52] Bowyer, Kevin W. et al. “SMOTE: Synthetic Minority Over-Sampling Technique.”, Journal of Artificial Intelligence Research, Volume 16, 321-357(2002).

[53] https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes (05.10.2018)

[54] https://scikit-learn.org/stable/modules/tree.html (05.10.2018)

[55] https://scikit-learn.org/stable/modules/svm.html (05.10.2018)

[56] https://community.alteryx.com/t5/Alteryx-Knowledge-Base/Seeing-the-Forest-for-the-Trees-An-Introduction-to-Random-Forest/ta-p/158062 (05.10.2018)